Софийски университет „Св. Климент Охридски”
Факултет по математика и информатика

Дипломна работа
на тема

Построяване на f-преобразувател на ниво символи за представяне на езикови модели

Георги Шопов

Научен ръководител
доц. Стоян Михов

28 март 2019г.

Съдържание

Увод

Езиков модел е вероятностна мярка върху множество от езикови единици. Ние ще се фокусираме върху случая, в който езиковите единици са изречения. Тоест езиковите модели приписват вероятност на всяко изречение. На практика езиковите модели се построяват (научават) от корпус – крайна редица от наблюдавани езикови единици (в случая изречения). Това прави производителността на езиковия модел пряко свързана с корпуса, от който е научен. Ако дадено изречение не се среща в корпуса, то вероятността, която построен от този корпус езиков модел ще припише на това изречение, би била много несигурна. В глава 1 описваме този проблем и един добре известен опит за неговото решаване – изгладените n-грамни езикови модели. Част от детайлите на описанието в глава 1 отделяме в приложение Б, което от своя страна се възползва от понятията и свойствата представени в приложение А. В останалите глави разглеждаме задачата за представяне специфично на изгладени n-грамни езикови модели.

Езиковите модели се използват в много задачи свързани с обработка на естествен език. Това прави задачата за ефективното им представяне стойностна. В настоящата дипломна работа ще разгледаме чисто автоматни представяния на изгладени n-грамни езикови модели. От класическата теория на крайните автомати знаем, че от гледна точка на времева и пространствена ефективност е удачно абстрактните машини, с които ще представяме езикови модели, да бъдат детерминистични и да могат ефективно да бъдат „минимизирани". Изгладените n-грамни езикови модели водят естествено до представяне с преобразователи – детерминистични машини, които разширяват понятието краен детерминиран автомат, позволявайки всеки преход да притежава не само входен символ, но и изход. В глава 2 разглеждаме конструкция на преобразувател, представящ изгладен n-грамен езиков модел.

Въпреки оптималността си откъм време за траверсиране, представянето на изгладен n-грамен езиков модел чрез преобразувател може да бъде подобрено от гледна точка на пространствена сложност. За целта в глава 3 разглеждаме абстрактни машини, които могат да се възползват в по-голяма степен от спецификата на изгладените n-грамни езикови модели. Тези машини наричаме f-преобразуватели – преобразуватели, които позволяват използването на специален вид преходи, които наричаме f-преходи. Allauzen, Mohri и Roark 2003 показват конструкция на f-преобразувател, представящ изгладен n-грамен езиков модел, чиято пространствена сложност е по-ниска (с фактор големината на входната азбука) от тази на представянето с преобразувател. Съществото при тази конструкция е, че подобрението в размера на представянето не е за сметка на времето за траверсиране – подобно на преобразувателите, конструираният f-преобразувател може да бъде траверсиран за линейно време относно дължината на входа. В глава 3 разглеждаме тази конструкция, някои нейни свойства и доказваме нейната коректност, тъй като доказателството е спестено в оригиналната статия.

В първите три глави гледаме на изреченията като на последователности от думи от даден речник, а на думите от речника като на неделими единици (символи). Тоест не се

интересуваме от тяхната структура. В глава 4 разглеждаме въпроса за представяне на изгладен n-грамен езиков модел чрез f-преобразувател, за който думите от изреченията са предварително изписани посредством някаква функция, тоест входът на f-преобразувателя представлява конкатенацията от изписванията на думите в дадено изречение. Конструкцията, която представяме, се базира на конструкцията от глава 3 и запазва свойството, че резултатният f-преобразувател може да бъде траверсиран за линейно време относно дължината на входа.

Представянията на изгладени n-грамни езикови модели, които разглеждаме в първите четири глави са оптимални откъм време за намиране на вероятността на дадено изречение. Неразгледан остава въпросът за тяхната минимизация, тоест въпросът за намиране на минималния относно брой състояния плюс брой преходи преобразувател (f-преобразувател), еквивалентен на изходния. Стандартният метод за минимизация на преобразуватели изисква първо те да бъдат канонизирани. Поради тази причина в глава 5 първо представяме теорията за канонизация на преобразуватели с изходи от моноида $([0,1],\cdot,1)$. След това показваме как канонизацията на f-преобразуватели, чиито подлежащи преобразуватели са с изходи от моноида $([0,1],\cdot,1)$, се свежда до канонизацията на подлежащите им преобразуватели. Тъй като f-преобразувателите на ниво думи, представящи изгладени n-грамни езикови модели, удовлетворяват това условие, по този начин получаваме метод за тяхното канонизиране. След това показваме, че специфичната структура на f-преобразувателите на ниво думи за изгладени n-грамни езикови модели с изходи от моноида $([0,1],\cdot,1)$, позволява те да бъдат канонизирани по-ефективно. Накрая показваме как могат да бъдат канонизирани f-преобразуватели на ниво символи за изгладени n-грамни езикови модели, независимо от това че подлежащите им преобразуватели могат да имат изходи по-големи от 1.

От класическата теория на крайните автомати знаем, че крайните детерминирани автомати могат да бъдат минимизирани ефективно. Също така знаем, че минимизацията на канонични преобразуватели, се свежда ефективно до задачата за минимизация на краен детерминиран автомат. В случая на f-преобразуватели нещата не стоят така. Задачата за минимизация на f-преобразувател е NP-пълна (следва от резултатите, описани в Björklund, Björklund и Zechner 2014). Поради тази причина в глава 6 отслабваме свойството за минималност и дефинираме понятието „псевдоминимален" f-преобразувател. Псевдоминималният f-преобразувател има свойството, че е не по-голям от изходния и може да бъде построен ефективно. Изложението в глава 6 е следното. Първо, представяме накратко част от теорията, свързана с минимизация на крайни детерминирани автомати и канонични преобразуватели, и описваме алгоритъма на Béal и Crochemore (Béal и Crochemore 2008) за минимизация на краен детерминиран автомат. След това дефинираме понятието „псевдоминимален" f-преобразувател и показваме как псевдоминимизацията на каноничен f-преобразувател се свежда до минимизацията на каноничен преобразувател. Тъй като минимизацията на каноничен преобразувател се свежда до минимизацията на краен детерминиран автомат, можем да използваме алгоритъма на Béal и Crochemore, за да псевдоминимизираме канонични f-преобразуватели. Накрая на главата показваме как можем да модифицираме алгоритъма на Béal и Crochemore така, че да подобрим пространствената му сложност в случаите, в които той се използва за псевдоминимизация на каноничен f-преобразувател, представящ езиков модел.

1 Езикови модели

Езиков модел е вероятностна мярка върху множество от езикови единици. Примери за такива езикови единици са думите и изреченията в естествените езици. Ние ще се фокусираме върху случая, в който езиковите единици са изречения. Тогава езиковите модели приписват мярка (вероятност) на всяко изречение.

Езиковите модели се използват в много задачи свързани с обработка на естествен език. Например при разпознаване на реч задачата е да се определи последователността от изказани думи в даден звуков сигнал. Част от трудностите, които възникват в тази задача, са свързани с нееднозначността между последователностите от думи и техните произношения. Например

- думи като „шев" и „шеф" са омофони – произнасят се по един и същ начин, но имат различен правопис и значение;
- словосъчетания като „всичките съвременни" и „всички те са временни" са граматически правилни, различни са по значение и правопис, но имат сходно произношение;
- словосъчетания като „яли меса пак" и „яли ме са пак" са различни по значение и правопис, но имат сходно произношение и единствено първото е граматически правилно.

Езиковите модели са едно възможно решение на проблема с тези многозначности. Вероятностите, които те приписват на всяко изречение, могат да бъдат използвани, за да бъде избрана най-вероятната последователност от думи. Тоест последователността от думи, която е в някакъв смисъл граматически „най-правилна" и чиито думи се съчетават в някакъв смисъл „най-добре" семантично.

Да отбележим, че изборът ни да моделираме естествен език по този начин – вероятностно, се основава на убедеността ни в съществуването на такава вероятностна мярка, описваща разглеждания феномен.

1.1 Основни понятия

Формално вероятностна мярка се дефинира върху σ-алгебра над дадено множество (в случая множество от изречения). Елементите на множеството (от изречения) наричаме елементарни събития, а елементите на σ-алгебрата – събития. Вероятностна мярка заедно с нейната σ-алгебра и множество от елементарни събития образуват вероятностно пространство.

Определение 1.1. *Вероятностно пространство* е тройка $(\Omega, \mathcal{A}, P)$, където

- Ω е непразно множество от елементарни събития;
- $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ е *σ-алгебра* над Ω, тоест

- $\Omega \in \mathcal{A}$;
- $\mathcal{A}$ е затворено относно допълнение до Ω, тоест ако $A \in \mathcal{A}$, то $\Omega \setminus A \in \mathcal{A}$;
- $\mathcal{A}$ е затворено относно изброимо обединение, тоест ако $(A_i)_{i=0}^{\infty}$ е редица от елементи на $\mathcal{A}$, то $\bigcup_{i=0}^{\infty} A_i \in \mathcal{A}$;

- $P\colon \mathcal{A} \to [0,1]$ е *вероятностна мярка*, тоест
  - $P(\Omega) = 1$;
  - P е *изброимо адитивна*, тоест ако $(A_i)_{i=0}^{\infty}$ е редица от два по два чужди елемента на $\mathcal{A}$, то $P(\bigcup_{i=0}^{\infty} A_i) = \sum_{i=0}^{\infty} P(A_i)$.

В настоящата дипломна работа ще правим разлика между термините „буква", „азбука", „дума" и „език" от теория на формалните езици и съответните им значения в контекста на естествените езици. Затова сега ще дадем формална дефиниция за всеки от тези термини.

Определение 1.2. *Азбука* наричаме всяко крайно множество. Всеки елемент на дадена азбука наричаме *буква* или *символ*.

Определение 1.3. *Дума над азбуката* Σ наричаме всяка крайна редица от букви от Σ. Дължината на дума α над Σ ще означаваме с $|\alpha|$. Единствената дума с дължина 0 ще наричаме *празната дума* и ще я бележим с ε.

Определение 1.4. Нека Σ е азбука и α и β са думи над Σ. Тогава *конкатенацията* на α и β ще наричаме думата $\alpha\beta$.

Определение 1.5. Нека Σ е азбука. Със $\Sigma^\star$ ще означаваме множеството от всички думи над Σ.

Определение 1.6. *Език над азбуката* Σ наричаме всяко подмножество на $\Sigma^\star$.

Под езиков модел ще разбираме вероятностно пространство (по-надолу ще се убедим, че можем да си мислим за езиковите модели дори само като за вероятностни мерки), чиито елементарни събития са всички изречения, съставени от думи от даден речник, и чиято вероятностна мярка е дефинирана за всяко изречение. Под речник ще разбираме крайно множество от думи, тоест азбука. Дума от даден речник представлява дума над дадена азбука. Засега азбуката, над която са думите от речника, е без значение, тоест на думите от речника ще гледаме като на неделими единици – букви. Под изречение, съставено от думи от даден речник, ще разбираме крайна редица от думи от съответния речник или еквивалентно – дума над азбуката, определена от речника. Така формално в дефинициите и твърденията ще говорим за букви от азбука и думи над азбука, а неформално за думи от речник и изречения, съставени от думи от речник. Многозначностите, които възникват при използването на термина „дума" – дума над азбука или дума от речник, ще бъдат разрешавани от контекста.

Една конвенция, която ще следваме, е, че с малки латински букви (евентуално с индекси) ще означаваме букви от дадена азбука (или думи от даден речник), а с малки гръцки букви (евентуално с индекси) ще означаваме думи над дадена азбука (или изречения, съставени от думи от даден речник).

Определение 1.7. Нека Σ е азбука. *Езиков модел над* $\Sigma^\star$ наричаме вероятностно пространство от вида $(\Sigma^\star, \mathcal{P}(\Sigma^\star), P)$.

Чрез непосредствена проверка на условията от дефиницията на σ-алгебра, се вижда, че за кой да е речник Σ множеството $\mathcal{P}(\Sigma^\star)$ е σ-алгебра над $\Sigma^\star$, тоест $(\Sigma^\star, \mathcal{P}(\Sigma^\star), P)$ е вероятностно пространство за коя да е вероятностна мярка $P\colon \mathcal{P}(\Sigma^\star) \to [0,1]$. Също така условието, което искаме P да удовлетворява – да е дефинирана за всяко изречение, тоест за всеки синглетон $\{\alpha\}$ за $\alpha \in \Sigma^\star$, определя еднозначно σ-алгебрата на вероятностното пространство. Това е така, защото

- всички синглетони $\{\alpha\}$ за $\alpha \in \Sigma^\star$ трябва да принадлежат на $\mathrm{Dom}(P)$, а следователно и на σ-алгебрата;
- $\Sigma^\star$ е изброимо, тоест всяко негово подмножество е изброимо и може да се представи като изброимо обединение на синглетони;
- по дефиниция σ-алгебрата е затворена относно изброимо обединение и следователно трябва да съдържа всяко подмножество на $\Sigma^\star$.

Тоест в дефиницията на езиков модел над $\Sigma^\star$ нямаме свободата да изберем друга σ-алгебра, ако искаме вероятностната мярка да бъде дефинирана за всяко изречение.

За вероятностно пространство $(\Sigma^\star, \mathcal{P}(\Sigma^\star), P)$ речникът Σ и множествата $\Sigma^\star$ и $\mathcal{P}(\Sigma^\star)$ се подразбират от самата вероятностна мярка P. Затова ще си позволяваме да наричаме езиков модел над $\Sigma^\star$ всяка вероятностна мярка $P\colon \mathcal{P}(\Sigma^\star) \to [0,1]$. Когато речникът Σ е без значение, ще използваме термина „езиков модел" вместо термина „езиков модел над $\Sigma^\star$".

За дискретно вероятностно пространство от вида $(\Omega, \mathcal{P}(\Omega), P)$, тоест такова, че Ω е изброимо, е в сила, че вероятностната мярка P се определя еднозначно от стойностите ѝ върху синглетоните $\{\omega\}$ за $\omega \in \Omega$. Тъй като $\Sigma^\star$ е изброимо за кой да е речник Σ, езиков модел P над $\Sigma^\star$ се определя еднозначно от стойностите $P(\{\alpha\})$ за $\alpha \in \Sigma^\star$. Иначе казано, когато дефинираме езиков модел, ще е достатъчно да посочим вероятностите, които той съпоставя на всяко от изреченията.

1.2 Параметризиране на езикови модели

Нека Σ е речник и P е езиков модел над $\Sigma^\star$. Използвайки формулата за вероятност на сечение на събития $A_1, A_2, \ldots, A_n \in \mathcal{P}(\Sigma^\star)$

$$P(\bigcap_{i=1}^{n} A_i) = \prod_{i=1}^{n} P(A_i \mid \bigcap_{j=1}^{i-1} A_i),$$

където $P(\bigcap_{i=1}^{n-1} A_i) \neq 0$, можем да изразим вероятността, приписана на изречение $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \ldots a_n$ от езиковия модел P, така

$$\begin{aligned} P(\{\alpha\}) &= P(\{\beta \in \Sigma^\star \mid |\beta| = n\} \cap \bigcap_{i=1}^{n} B_i) \\ &= P(\{\beta \in \Sigma^\star \mid |\beta| = n\}) \prod_{i=1}^{n} P(B_i \mid \{\beta \in \Sigma^\star \mid |\beta| = n\} \cap \bigcap_{j=1}^{i-1} B_j) \\ &= P(\{\beta \in \Sigma^\star \mid |\beta| = n\}) \prod_{i=1}^{n} P(B_i \mid \{\beta \in \Sigma^\star \mid |\beta| = n\} \cap B_{i-1}), \end{aligned}$$

при положение че $P(\{\beta \in \Sigma^\star \mid |\beta| = n\} \cap B_{n-1}) \neq 0$ и за $i \in \{0, 1, \ldots, n\}$

$$B_i = \{\beta \in \Sigma^\star \mid a_1 a_2 \ldots a_i \preceq \beta\}.$$

Тук $\alpha \preceq \beta$ за $\alpha, \beta \in \Sigma^\star$ означава, че изречението α е префикс на изречението β.

Определение 1.8. Нека Σ е азбука. Казваме, че думата $\alpha \in \Sigma^\star$ е *префикс* на думата $\beta \in \Sigma^\star$, което означаваме с $\alpha \preceq \beta$, тогава и само тогава, когато $(\exists \gamma \in \Sigma^\star)(\alpha\gamma = \beta)$.

За да премахнем нуждата от събитието, фиксиращо дължината на изречението, можем да въведем нова дума $\$ \notin \Sigma$ и да разглеждаме езиковите модели над $\Sigma^\star \circ \{\$\}$ вместо над $\Sigma^\star$. Тук с $A \circ B$ означаваме конкатенацията на множествата от изречения A и B.

Определение 1.9. Нека Σ е азбука. Функцията $\circ\colon \mathcal{P}(\Sigma^\star) \times \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star)$ дефинираме за $A, B \in \mathcal{P}(\Sigma^\star)$ така

$$\circ(A, B) = \{\alpha\beta \mid \alpha \in A \wedge \beta \in B\}.$$

$\circ(A, B)$ наричаме *конкатенацията* на A и B и означаваме с $A \circ B$.

Така думата $ ще определя края на изречението. За удобство, което ще стане ясно по-надолу, ще въведем и друга нова дума $\hat{} \notin \Sigma \cup \{\$\}$, която да маркира началото на изречението. Така ще считаме, че думите $\hat{}$ и $ са различни и изначално фиксирани. Всички речници, за които изискваме да не съдържат новите думи ($\hat{}$ и $), ще отбелязваме, като поставяме вълна ($\widetilde{}$) над тях. Оттук нататък ще разглеждаме единствено езиковите модели над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, където $\widetilde{\Sigma}$ е произволен речник, несъдържащ $\hat{}$ и $. Фокусирането ни само върху тези езикови модели не ни ограничава, както се вижда от следното твърдение, което не доказваме, тъй като се съобразява лесно.

Твърдение 1.1. *Нека Σ е азбука и $\sigma_1, \sigma_2 \notin \Sigma$ са два различни нови символа. Тогава функцията $f\colon A \to B$, дефинирана за $P \in A$ така*

$$f(P) = Q \iff (\forall \alpha \in \Sigma^\star)(P(\{\alpha\}) = Q(\{\sigma_1\alpha\sigma_2\})),$$

е биекция между множествата

$$A = \{P \mid P\colon \mathcal{P}(\Sigma^\star) \to [0,1] \text{ е езиков модел над } \Sigma^\star\},$$
$$B = \{P \mid P\colon \mathcal{P}(\{\sigma_1\} \circ \Sigma^\star \circ \{\sigma_2\}) \to [0,1] \text{ е езиков модел над } \{\sigma_1\} \circ \Sigma^\star \circ \{\sigma_2\}\}. \qquad \square$$

Сега всяко събитие от вида $\{\alpha\}$, където $\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \ldots a_{n+2}$, може да се изрази като

$$\bigcap_{i=1}^{n+2} \{\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid a_1 a_2 \ldots a_i \preceq \beta\},$$

а вероятността, приписана му от езиков модел P, можем да изразим така

$$P(\{\alpha\}) = P(\bigcap_{i=1}^{n+2} B_i) = \prod_{i=1}^{n+2} P(B_i \mid \bigcap_{j=1}^{i-1} B_j) = \prod_{i=1}^{n+2} P(B_i \mid B_{i-1}) = \prod_{i=2}^{n+2} P(B_i \mid B_{i-1}),$$

при положение че $P(B_{n+1}) \neq 0$ и за $i \in \{0, 1, \ldots, n+2\}$

$$B_i = \{\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid a_1 a_2 \ldots a_i \preceq \beta\}.$$

Последното равенство следва от това, че

$$B_0 = \{\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid \varepsilon \preceq \beta\} = \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\},$$
$$B_1 = \{\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid \hat{} \preceq \beta\} = \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\},$$

$$P(B_1 \mid B_0) = P(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) = P(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) = 1.$$

Тъй като не знаем истинския езиков модел $P\colon \mathcal{P}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \to [0,1]$, който описва феномена и в чието съществуване вярваме, целта ни ще е да го приближим на базата на наблюдения. В случая наблюденията ще бъдат изречения, съставени от думи от даден речник. Последователността от наблюдения (изречения), с които разполагаме, ще наричаме корпус. Да забележим, че в даден корпус може да има повече от едно срещане на едно и също изречение.

Определение 1.10. Нека $\widetilde{\Sigma}$ е азбука. *Корпус над* $\widetilde{\Sigma}$ наричаме всеки елемент на

$$\bigcup_{n\geq 1} (\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})^n.$$

Използвайки корпус $\mathcal{C}$, бихме искали да научим само определени „параметри", от които еднозначно да можем да възстановим целия езиков модел. Такова свойство, както видяхме, имат условните вероятности от вида „вероятността следващата дума да е a при условие, че предходните думи са α", където αa е префикс на изречение от вероятностното пространство.

Определение 1.11. Нека $\widetilde{\Sigma}$ е азбука и P е езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$. *Параметър* на езиковия модел P наричаме всяка условна вероятност от вида

$$P(B_{\alpha a} \mid B_\alpha) \text{ за } \alpha a \in \mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \setminus \{\hat{}\},$$

където за $\alpha \in \mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})$

$$B_\alpha = \{\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\} \mid \alpha \preceq \beta\}.$$

С Pref означаваме функцията, която съпоставя на съвкупност от изречения множеството от всички техни префикси.

Определение 1.12. Нека Σ е азбука. Функцията $\mathrm{Pref}\colon \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star)$ дефинираме за $A \in \mathcal{P}(\Sigma^\star)$ така

$$\mathrm{Pref}(A) = \{\beta \in \Sigma^\star \mid (\exists \alpha \in A)(\beta \preceq \alpha)\}.$$

Намирайки параметрите на езиков модел P, можем да възстановим вероятностите, които той приписва, на всички синглетони, а оттам и на всички събития. Да отбележим, че $P(B_{\hat{}} \mid B_\varepsilon)$ не е параметър, тъй като, както вече видяхме, тази условна вероятност е винаги равна на 1. Важно е да припомним, че условните вероятности от вида $P(B_{\alpha a} \mid B_\alpha)$ са дефинирани само ако $P(B_\alpha) > 0$. Тоест всеки един от параметрите може да приеме стойност в интервала $[0,1]$ или да бъде недефиниран. Недефинираността на част от параметрите не променя факта, че можем да намерим вероятностите на всички синглетони. Ако параметърът $P(B_{\alpha a} \mid B_\alpha)$ е недефиниран и $\alpha a \preceq \beta$, то $P(\{\beta\}) = 0$, защото

$$\{\beta\} \subseteq B_\alpha \implies P(\{\beta\}) \leq P(B_\alpha) = 0.$$

1.3 n-грамни езикови модели

Съществена пречка при научаването на параметрите на езиков модел P е, че те са безброй много. За да ограничим техния брой, можем да предположим, че езиковият модел P, който търсим (приближаваме), притежава свойството на Марков от ред $n \in \mathbb{N}^+$. Тоест предполагаме, че вероятността следващата дума да е a зависи единствено от последните не повече от $n-1$ думи, видени преди нея. Тези $n-1$ думи, предшестващи a, ще наричаме история на a.

Определение 1.13. Нека Σ е азбука и $n \in \mathbb{N}^+$. Дефинираме функцията $h_n \colon \Sigma^\star \to \Sigma^\star$ за $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \dots a_m$ така

$$h_n(\alpha) = a_{\max\{1, m-n+2\}} \dots a_m.$$

Определение 1.14. Нека $\widetilde{\Sigma}$ е азбука и $n \in \mathbb{N}^+$. Езиковия модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ наричаме *n-грамен езиков модел над* $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, ако P притежава *свойството на Марков от ред* n, тоест ако за всеки $\alpha a, \beta a \in \mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \setminus \{\hat{}\}$ е в сила, че

$$P(B_\alpha) > 0 \wedge P(B_\beta) > 0 \wedge h_n(\alpha) = h_n(\beta) \implies P(B_{\alpha a} \mid B_\alpha) = P(B_{\beta a} \mid B_\beta).$$

Числото n наричаме *ред* на езиковия модел P.

Често ще използваме термина „n-грамен езиков модел" вместо „n-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$", когато речникът $\widetilde{\Sigma}$ е без значение. От предходната дефиниция можем да заключим, че за даден n-грамен езиков модел е достатъчно да знаем параметрите с различни истории и че техният брой е не по-голям от броя на различните изречения $h_n(\alpha)a$ за $\alpha a \in \mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \setminus \{\hat{}\}$. Тоест параметрите, от които се интересуваме, са от порядъка на

$$\sum_{i=0}^{n} (|\widetilde{\Sigma}| + 2)^i = \frac{(|\widetilde{\Sigma}| + 2)^{n+1} - 1}{(|\widetilde{\Sigma}| + 2) - 1} \in O(|\widetilde{\Sigma}|^n).$$

С цел отразяване на забележката, че се нуждаем само от параметрите с различни истории, и с цел изразяване на вероятността на произволно изречение по единен начин, [1] ще въведем означение, с което ще отбелязваме стойностите на параметрите с различни истории на n-грамен езиков модел, като ще считаме, че недефинираните параметри имат стойност произволно число в интервала $[0, 1]$.

В следващата дефиницията се възползваме от това, че изречение αa принадлежи на $\mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \setminus \{\hat{}\}$ тогава и само тогава, когато a принадлежи на $\widetilde{\Sigma} \cup \{\$\}$ и α принадлежи на $\{\hat{}\} \circ \widetilde{\Sigma}^\star$.

Определение 1.15. Нека $\widetilde{\Sigma}$ е азбука и P е n-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$. С Ξ_P означаваме множеството от всички функции от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0, 1]$, такова че всяка $\xi \in \Xi_P$ е дефинирана за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ така

$$\xi(a, \alpha) = \begin{cases} P(B_{\beta a} \mid B_\beta), & \text{ако } !\, P(B_{\beta a} \mid B_\beta) \\ r^\xi_{\alpha a}, & \text{иначе} \end{cases}$$

където $r^\xi_{\alpha a} \in [0, 1]$ и $\beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star$ е такова, че $h_n(\beta) = \alpha$.

Коректността на дефиницията на функция $\xi \in \Xi_P$ за n-грамен езиков модел P следва от свойството на Марков от ред n, което n-грамните езикови модели притежават. Оттук нататък за n-грамен езиков модел P с $P(a \mid \alpha)$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ ще означаваме стойностите $\xi(a, \alpha)$ на някоя функция $\xi \in \Xi_P$. Ще покажем, че изборът на функцията $\xi \in \Xi_P$, чиито стойности $\xi(a, \alpha)$ означаваме с $P(a \mid \alpha)$, е без значение, тъй като всяка функция $\xi \in \Xi_P$ определя еднозначно езиковия модел P.

1. Досега не можехме да изразяваме вероятността на изречение αa като произведение на параметри, когато вероятността на B_α е 0.

Твърдение 1.2. *Нека $\widetilde{\Sigma}$ е азбука и P е n-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$. Тогава за всяко $\xi \in \Xi_P$ и $\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \dots a_{m+2}$ е в сила, че*

$$P(\{\alpha\}) = \prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1})).$$

Доказателство. Ако α е такова, че $P(B_{a_1 a_2 \dots a_{m+1}}) \neq 0$, то всички параметри $P(B_{a_1 a_2 \dots a_i} \mid B_{a_1 a_2 \dots a_{i-1}})$ ще бъдат дефинирани и равни на съответните стойности $\xi(a_i, h_n(a_1 a_2 \dots a_{i-1}))$. Тоест

$$P(\{\alpha\}) = \prod_{i=2}^{m+2} P(B_{a_1 a_2 \dots a_i} \mid B_{a_1 a_2 \dots a_{i-1}}) = \prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1})).$$

В противен случай, ако $P(B_{a_1 a_2 \dots a_{m+1}}) = 0$, то $P(\{\alpha\}) = 0$ и съществува префикс $\beta a \preceq \alpha$, такъв че $P(B_\beta) = 0$. Нека βa е най-късият префикс на α, такъв че $P(B_\beta) = 0$. Ясно е, че $|\beta| > 0$, тъй като $P(B_\varepsilon) = 1$. Тоест $\beta = \beta' b$. Нещо повече, щом $P(B_{\beta' b}) = 0$ и $P(B_{\beta'}) \neq 0$, то

$$\xi(b, h_n(\beta')) = P(B_{\beta' b} \mid B_{\beta'}) = \frac{P(B_{\beta' b})}{P(B_{\beta'})} = 0.$$

Следователно

$$P(\{\alpha\}) = \prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1})).$$ □

Тъй като параметрите на един n-грамен езиков модел и стойностите $P(a \mid \alpha)$ носят една и съща информация (определят еднозначно езиковия модел), ще използваме термина „параметър" и за двете. Когато казваме, че n-грамен езиков модел P има параметри $P(a \mid \alpha)$, ще имаме предвид, че съществува функция $\xi\colon (\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\} \to [0,1]$ със стойности $\xi(a, \alpha) = P(a \mid \alpha)$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, която принадлежи на Ξ_P. Когато използваме израза „да научим параметрите $P(a \mid \alpha)$ на n-грамен езиков модел P", ще имаме предвид да намерим функция $\xi \in \Xi_P$, чиито стойности да бъдат отразени от параметрите $P(a \mid \alpha)$.

Както вече споменахме, целта ни е да индуцираме езиков модел от даден корпус. Също така отбелязахме, че това е еквивалентно на това да научим неговите параметри. Тоест целим да намерим функция $\xi\colon (\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\} \to [0,1]$, която определя n-грамен езиков модел, тоест за която съществува n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$. Сега ще формулираме необходимо и достатъчно условие такава функция да определя n-грамен езиков модел по горепосочения начин. Доказателството ще представим в приложение Б, като в него се използват съществено понятия и свойства, описани в приложение А.

Твърдение 1.3. *Нека $\widetilde{\Sigma}$ е азбука, $n \in \mathbb{N}^+$ и ξ е функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0,1]$. Тогава съществува n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$, тогава и само тогава, когато*

- $(\exists a_1 a_2 \dots a_{m+2} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})(\prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1}) > 0)$;
- $(\forall \alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\})(\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \alpha) = 1)$. □

Едни параметри $P(a \mid \alpha)$, които могат да бъдат научени от корпус $\mathcal{C}$, са параметрите, определящи езиковия модел, максимизиращ правдоподобието на корпуса $\mathcal{C}$.

Определение 1.16. Нека $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ е корпус над азбуката $\widetilde{\Sigma}$. *Правдоподобие* на $\mathcal{C}$ относно езиков модел P над $\{\^{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ наричаме стойността

$$\prod_{i=1}^{n} P(\{\alpha_i\}).$$

Определение 1.17. Нека Σ е азбука и $\alpha = a_1 a_2 \ldots a_n$ е дума над Σ. Функцията $\#_\alpha \colon \Sigma^\star \to \mathbb{N}$ дефинираме за $\beta \in \Sigma^\star$ така

$$\#_\alpha(\beta) = \begin{cases} |\alpha| + 1, & \text{ако } \beta = \varepsilon \\ |\{(i,j) \mid 1 \leq i \leq j \leq n \wedge a_i a_{i+1} \ldots a_j = \beta\}|, & \text{иначе} \end{cases}$$

Определение 1.18. Нека $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ е корпус над азбуката $\widetilde{\Sigma}$. Функцията $\#_\mathcal{C} \colon (\widetilde{\Sigma} \cup \{\^{}, \$\})^\star \to \mathbb{N}$ дефинираме за $\alpha \in (\widetilde{\Sigma} \cup \{\^{}, \$\})^\star$ така

$$\#_\mathcal{C}(\alpha) = \sum_{i=1}^{n} \#_{\alpha_i}(\alpha).$$

С цел облекчаване на записа с $\#_\mathcal{C}(\alpha\bullet)$ за $\alpha \in (\widetilde{\Sigma} \cup \{\^{}, \$\})^\star$ ще означаваме сумата

$$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \#_\mathcal{C}(\alpha a).$$

Определение 1.19. Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $n \in \mathbb{N}^+$. *n-грамен езиков модел, максимизиращ правдоподобието на корпуса $\mathcal{C}$* наричаме n-грамния езиков модел над $\{\^{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, чиито параметри $P(a \mid \alpha)$ са дефинирани за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\^{}\} \circ \widetilde{\Sigma}^\star\}$ така

$$P(a \mid \alpha) = \begin{cases} \frac{\#_\mathcal{C}(\alpha a)}{\#_\mathcal{C}(\alpha\bullet)}, & \text{ако } \#_\mathcal{C}(\alpha a) > 0 \\ 0, & \text{иначе} \end{cases}$$

Тук се възползвахме от нововъведената дума $\^{}$, която ни позволява да говорим за началата на изреченията. Благодарение на нея успяхме да изразим всички параметри на n-грамния езиков модел, максимизиращ правдоподобието на даден корпус, по единен начин. Ако не я бяхме въвели, в случаите, когато α е префикс с ненулева дължина на изречение от вероятностното пространство, щеше да се наложи да уточняваме, че в частното $\frac{\#_\mathcal{C}(\alpha a)}{\#_\mathcal{C}(\alpha\bullet)}$ искаме да броим само срещанията, които са в началото на изречение.

Сега ще покажем, че стойностите $P(a \mid \alpha)$ от определение 1.19 са наистина параметри на n-грамен езиков модел.

Твърдение 1.4. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $n \in \mathbb{N}^+$. Нека $P(a \mid \alpha)$ са стойностите от определение 1.19 за корпуса $\mathcal{C}$ и азбуката $\widetilde{\Sigma}$. Тогава съществува n-грамен езиков модел P над $\{\^{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ с параметри $P(a \mid \alpha)$.*

Доказателство. От дефиницията на корпус следва, че $\mathcal{C}$ съдържа поне едно изречение $\alpha \in \{\^{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$. Нека $\alpha = a_1 a_2 \ldots a_{m+2}$ е изречение от корпуса $\mathcal{C}$. Тогава

$$(\forall 1 \leq i \leq m+2)(\#_\mathcal{C}(h_n(a_1 a_2 \ldots a_{i-1}) a_i) > 0),$$

откъдето следва, че

$$\prod_{i=2}^{m+2} P(a_i \mid h_n(a_1 a_2 \ldots a_{i-1})) = \prod_{i=2}^{m+2} \frac{\#_\mathcal{C}(h_n(a_1 a_2 \ldots a_{i-1}) a_i)}{\#_\mathcal{C}(h_n(a_1 a_2 \ldots a_{i-1})\bullet)} > 0.$$

Нека сега $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$. Тогава

$$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P(a \mid \alpha) = \sum_{\substack{a \in \widetilde{\Sigma} \cup \{\$\}: \\ \#_{\mathcal{C}}(\alpha a) > 0}} P(a \mid \alpha) = \sum_{\substack{a \in \widetilde{\Sigma} \cup \{\$\}: \\ \#_{\mathcal{C}}(\alpha a) > 0}} \frac{\#_{\mathcal{C}}(\alpha a)}{\#_{\mathcal{C}}(\alpha \bullet)} = \frac{\sum_{\substack{a \in \widetilde{\Sigma} \cup \{\$\}: \\ \#_{\mathcal{C}}(\alpha a) > 0}} \#_{\mathcal{C}}(\alpha a)}{\#_{\mathcal{C}}(\alpha \bullet)} = 1.$$

Сега от твърдение 1.3 следва, че съществува n-грамен езиков модел P с параметри $P(a \mid \alpha)$. □

Доказателство, че n-грамният езиков модел от определение 1.19 максимизира правдоподобието на корпуса $\mathcal{C}$, дават Ney, Martin и Wessel 1997.

1.4 Изгладени n-грамни езикови модели

Дори след въвеждането на n-грамните езикови модели с цел намаляване на броя параметри, които трябва да бъдат научени, броят параметри продължава да зависи експоненциално от n. Например дори за малък речник ($|\widetilde{\Sigma}| = 30000$) броят параметри експлодира, както се вижда от таблица 1.1.

Модел	**Параметри**
$n = 1$	$\sim$ 30 хиляди
$n = 2$	$\sim$ 900 милиона
$n = 3$	$\sim$ 27 трилиона

Таблица 1.1: Брой параметри при различни n и речник с 30 хиляди думи.

С увеличаването на броя параметри нараства и количеството изречения в корпуса, от които се нуждаем, за да можем надеждно да научим параметрите. В горния пример при $n = 3$ бихме се нуждаели от корпус с поне 27 трилиона думи, за да е теоретично възможно да сме срещнали поне по един свидетел в корпуса за всеки от параметрите. На практика обаче бихме имали нужда от много по-голям корпус, тъй като някои последователности от думи се срещат по-рядко от други. [2] Разбира се, когато строим езиков модел, корпусът, с който разполагаме, е предварително фиксиран – обичайно той е максималният брой изречения от съответната област, с които сме успели да се снабдим. Фиксиран е и речникът – той представлява множеството от използваните в областта, в която ще прилагаме езиковия модел, думи. В този случай дори за малки n може да има много последователности от думи с дължина не по-голяма от n, които не са се срещнали в корпуса или са се срещнали, но само по случайност. Приближенията, които бихме получили за параметрите, съответстващи на тези последователности, биха били много несигурни. Поради тази причина се налага

- или да използваме езиков модел от по-нисък ред, като по този начин да намалим броя на параметрите с надеждата, че ще успеем „по-добре" да се справим с научаването на по-малък брой параметри;
- или да „изгладим" n-грамния езиков модел – да преразпределим вероятностната маса във вероятностното пространство така, че да поправим неточностите, възникнали поради недостатъчното количество и разнообразие на изречения в корпуса (например

2. Емпиричният закон на Zipf гласи, че за корпус съставен от изречения на естествен език е в сила, че i-тата най-често срещана дума има честота (брой срещания) обратнопропорционална на i.

да пренесем вероятностна маса от параметрите с високи стойности върху параметрите с ниски стойности, така че да няма нулеви параметри, тоест да няма изречения с нулева вероятност).

Използването само на първия подход има съществени недостатъци. От една страна, при определени речници $\widetilde{\Sigma}$ и определени корпуси $\mathcal{C}$ над $\widetilde{\Sigma}$ винаги ще има параметри, за които няма свидетел в корпуса независимо от това колко малко n изберем. От друга страна, с намаляването на n спада дискриминативността на модела. Например при $n = 1$ вероятността на изречението „при съвпадение на книжовна и диалектна дума" би била една и съща с вероятността на коя да е пермутация на неговите думи, като например „диалектна съвпадение дума при книжовна и", което е силно нежелан ефект най-малкото, защото второто изречение не е граматически правилно. Също така можем да забележим, че намалявайки n, ограничаваме възможността на модела да отразява далечни зависимости между думи. Например в изречението „момчето прескочи голямата зелена ___" глаголът „прескочи" има силно влияние върху това коя ще е думата след „зелена". При 4-грамен езиков модел историята, с която разполагаме, когато смятаме вероятността следващата дума да е „ограда", ще бъде с дължина 3 и ще обхване думата „прескочи". Ако n е 3, историята на „ограда" би била само „голяма зелена", но в този контекст „ограда" не е най-естествената дума. Други думи, като „ябълка" или „планина", са много по-често срещани след „голяма зелена", въпреки неуместността им в конкретното изречение, дължаща се на глагола „прескочи". Тоест, използвайки по-малко n, ние намаляваме точността на езиковия модел.

Поради тази причина се използват и двата подхода. Подходящото n (редът на езиковия модел) се избира спрямо големината на корпуса и след това езиковият модел се изглажда, като целта е не само избягване на нулевите параметри, но и подобряване на точността на модела като цяло. В тази връзка Chen и Goodman 1998 казват следното.

> Whenever data sparsity is an issue, smoothing can help performance, and data sparsity is almost always an issue in statistical modeling. In the extreme case where there is so much training data that all parameters can be accurately trained without smoothing, one can almost always expand the model, such as by moving to a higher-order n-gram model, to achieve improved performance. With more parameters data sparsity becomes an issue again, but with proper smoothing the models are usually more accurate than the original models. Thus, no matter how much data one has, smoothing can almost always help performance, and for a relatively small effort. [3]

Kneser и Ney 1995 отбелязват (Chen и Goodman 1998 също наблягат на това наблюдение), че почти всички техники за изглаждане на n-грамни езикови модели, научени от корпус, могат да бъдат описани по единен начин. По-конкретно, параметрите на почти всеки изгладен n-грамен езиков модел $P^{(n)}$ за корпус $\mathcal{C}$ могат да бъдат представени така

$$P^{(n)}(a \mid \alpha) = \begin{cases} d_{\alpha a}, & \text{ако } \#_{\mathcal{C}}(\alpha a) > 0 \\ e_{\alpha} P^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases}$$

3. Винаги когато оскъдицата на данни е проблем, изглаждането на езиковия модел може да подобри неговата производителност, а в статистическото моделиране данните винаги са в оскъдица. Дори в крайни случаи, когато има достатъчно данни, така че всички параметри да бъдат научени точно без изглаждане, моделът може почти винаги да бъде разширен – например чрез увеличаване на неговия ред, и по този начин да бъде постигната по-добра производителност. С увеличаването на реда на модела се увеличава броят на параметрите и недостигът на данни отново става проблем, но с подходящо изглаждане обикновено получените модели са по-точни от изходните. Затова, независимо от това с какво количество данни разполага човек, изглаждащите техники могат почти винаги да подобрят производителността на модела и то за сметка на сравнително малко усилия.

където $P^{(n-1)}$ е изгладен $(n-1)$-грамен езиков модел за $\mathcal{C}$, $\alpha' \succeq \alpha$, $|\alpha'| = |\alpha|-1$ и $d_{\alpha a}, e_\alpha \in \mathbb{R}_{\geq 0}$ са такива, че $\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P^{(n)}(a \mid \alpha) = 1$. Тук $\alpha \succeq \beta$ означава, че изречението α е суфикс на изречението β.

Определение 1.20. Нека Σ е азбука. Казваме, че думата $\alpha \in \Sigma^\star$ е *суфикс* на думата $\beta \in \Sigma^\star$ и го означаваме с $\alpha \succeq \beta$ тогава и само тогава, когато $(\exists \gamma \in \Sigma^\star)(\gamma\alpha = \beta)$.

Разбира се, при $\alpha = \varepsilon$ не съществува $\alpha' \succeq \alpha$, такова че $|\alpha'| = |\alpha| - 1$. В този случай позволяваме на мястото на $P^{(0)}(a \mid \alpha')$ да стои произволно неотрицателно реално число. Това е еквивалентно на това да отъждествим изгладените 1-грамни езикови модели за $\mathcal{C}$ с 1-грамните езикови модели, тъй като даваме пълна свобода при избора на стойностите на параметрите.

Горепосоченото представяне важи само за параметрите, за които $\#_{\mathcal{C}}(\alpha) > 0$. За останалите параметри $P^{(n)}(a \mid \alpha)$ се отъждествява с $P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha))$, където $g_{\mathcal{C}}(\alpha)$ е най-дългият суфикс на α, който се среща в корпуса $\mathcal{C}$.

Определение 1.21. Нека $\widetilde{\Sigma}$ е азбука и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. Дефинираме функцията $g_{\mathcal{C}} \colon (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \to (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star$ за $\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star$, $\alpha = a_1 a_2 \dots a_n$ така

$$g_{\mathcal{C}}(\alpha) = \begin{cases} \varepsilon, & \text{ако } \alpha = \varepsilon \\ a_{\min\{1 \leq i \leq n \mid \#_{\mathcal{C}}(a_i a_{i+1} \dots a_n) > 0\}} \dots a_n, & \text{иначе} \end{cases}$$

Целта на това отъждествяване е за $P^{(n)}(a \mid \alpha)$ да се избере стойността, дадена от езиковия модел от най-висок ред („най-точния" езиков модел), който е видял част от историята α на този параметър (тоест има смислено негово приближение).

Тъй като най-често използваните изглаждащи техники могат да се изразят по горепосочения начин, [4] ще използваме това представяне като дефиниция за изгладен n-грамен езиков модел.

Определение 1.22. Нека $\widetilde{\Sigma}$ е азбука, $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$ и $n \in \mathbb{N}^+$. *Изгладен n-грамен езиков модел $P^{(n)}$ за корпуса $\mathcal{C}$* дефинираме индуктивно

- $P^{(1)}$ е изгладен 1-грамен езиков модел за $\mathcal{C}$, ако $P^{(1)}$ е 1-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, чиито параметри са дефинирани за $a \in \widetilde{\Sigma} \cup \{\$\}$ така

  $$P^{(1)}(a \mid \varepsilon) = d_a,$$

  където $d_a \in \mathbb{R}_{\geq 0}$ са такива, че

  $$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P^{(1)}(a \mid \varepsilon) = 1.$$

- $P^{(n+1)}$ е изгладен $(n+1)$-грамен езиков модел за $\mathcal{C}$, ако $P^{(n+1)}$ е $(n+1)$-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, чиито параметри са дефинирани за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_{n+1}(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ така

  $$P^{(n+1)}(a \mid \alpha) = \begin{cases} P^{(|\beta|+1)}(a \mid \beta), & \text{ако } |\beta| < n \\ d_{\alpha a}, & \text{ако } |\beta| = n \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P^{(n)}(a \mid \alpha'), & \text{иначе} \end{cases}$$

4. Всички изглаждащи техники, имплементирани в инструмента за научаване и прилагане на езикови модели SRILM (Stolcke 2002, Stolcke и др. 2011), попадат в тази категория.

където $\beta = g_{\mathcal{C}}(\alpha)$, $\alpha' \succeq \alpha$, $|\alpha'| = |\alpha| - 1$ и $d_{\alpha a}, e_\alpha \in \mathbb{R}_{\geq 0}$ са такива, че

$$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P^{(n+1)}(a \mid \alpha) = 1.$$

От твърдение 1.3 следва, че независимо от избора на константите $d_{\alpha a}$ и e_α, стойностите $P^{(n)}(a \mid \alpha)$ от определение 1.22 са параметри на n-грамен езиков модел тогава и само тогава, когато съществува изречение, на което те приписват ненулева вероятност.

Твърдение 1.5. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $n \in \mathbb{N}^+$. Нека $P^{(n)}(a \mid \alpha)$ са стойностите от определение 1.22 за корпуса $\mathcal{C}$ и азбуката $\widetilde{\Sigma}$. Тогава съществува n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ с параметри $P^{(n)}(a \mid \alpha)$ тогава и само тогава, когато*

$$(\exists a_1 a_2 \dots a_{m+2} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})(\prod_{i=2}^{m+2} P^{(n)}(a_i \mid h_n(a_1 a_2 \dots a_{i-1})) > 0).$$

Доказателство. Правата посока следва директно от твърдение 1.3. За обратната посока от определение 1.22 следва, че за всяко $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$

$$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P^{(n)}(a \mid \alpha) = 1.$$

Следователно, отново съгласно твърдение 1.3, съществува n-грамен езиков модел P с параметри $P^{(n)}(a \mid \alpha)$. □

Оттук нататък при фиксиран изгладен n-грамен езиков модел $P^{(n)}$ за корпус $\mathcal{C}$ с $P^{(i)}$ за $1 \leq i \leq n-1$ ще означаваме точно изгладения i-грамен езиков модел за $\mathcal{C}$, който участва в дефиницията на $P^{(n)}$, а с $d_{\alpha a}$ и e_α ще означаваме неотрицателните реални числа от дефинициите на $P^{(1)}, P^{(2)}, \dots, P^{(n)}$. Също така ще използваме термина „изгладен n-грамен езиков модел", тоест няма да споменаваме корпуса, когато той е без значение.

От определение 1.22 можем да заключим, че всеки изгладен n-грамен езиков модел $P^{(n)}$ се определя еднозначно не само от всичките си параметри, но и от всичките параметри на $P^{(1)}$, заедно с параметрите на $P^{(2)}, P^{(3)}, \dots, P^{(n)}$, за които има представител в корпуса, и константите e_α.

Твърдение 1.6. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$. Нека $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава $P^{(n)}$ се определя еднозначно от функциите D_n и E_n, където $D_n \colon (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \to [0,1]$ е дефинирана така*

- *$D_n(a) = d_a$ за $a \in \widetilde{\Sigma} \cup \{\$\}$;*
- *$D_n(\alpha a) = d_{\alpha a}$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \bigcup_{i=2}^{n} \{h_i(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, такива че $\#_{\mathcal{C}}(\alpha a) > 0$,*

а $E_n \colon (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \to [0,1]$ е дефинирана така

- *$E_n(\alpha) = e_\alpha$ за $\alpha \in \bigcup_{i=2}^{n} \{h_i(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, такова че $\#_{\mathcal{C}}(\alpha) > 0$.*

Доказателство. Твърдението ще докажем с индукция по n.

$n = 1$
Тогава D_1 представя всички параметри на $P^{(1)}$, а $E_1 = \varnothing$ и следователно го определят еднозначно.

$\boldsymbol{n \rightsquigarrow n+1}$

Нека твърдението е вярно за всяко $1 \leq i \leq n$ и да разгледаме параметъра $P^{(n+1)}(a \mid \alpha)$. Ще докажем, че можем да изразим този параметър посредством функциите D_{n+1} и E_{n+1}.

Ако $|g_{\mathcal{C}}(\alpha)| < n$, то $P^{(n+1)}(a \mid \alpha) = P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha))$ и от индукционното предположение знаем, че $P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha))$ може да се изрази посредством функциите $D_{|g_{\mathcal{C}}(\alpha)|+1}$ и $E_{|g_{\mathcal{C}}(\alpha)|+1}$. Тъй като $D_{|g_{\mathcal{C}}(\alpha)|+1} \subseteq D_{n+1}$ и $E_{|g_{\mathcal{C}}(\alpha)|+1} \subseteq E_{n+1}$, доказахме, че $P^{(n+1)}(a \mid \alpha)$ може да се изрази чрез функциите D_{n+1} и E_{n+1}.

Ако $|g_{\mathcal{C}}(\alpha)| = n$, то в случай че $\#_{\mathcal{C}}(\alpha a) > 0$, $P^{(n+1)}(a \mid \alpha) = d_{\alpha a} = D_{n+1}(\alpha a)$. В противен случай $P^{(n+1)}(a \mid \alpha) = e_\alpha P^{(n)}(a \mid \alpha')$, където $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$. От индукционното предположение знаем, че $P^{(n)}(a \mid \alpha')$ може да се изрази посредством функциите $D_n \subseteq D_{n+1}$ и $E_n \subseteq E_{n+1}$. Също така $E_{n+1}(\alpha) = e_\alpha$, следователно $P^{(n+1)}(a \mid \alpha)$ може да се изрази чрез функциите D_{n+1} и E_{n+1}. □

1.5 Примери за изгладени n-грамни езикови модели

Нека $\widetilde{\Sigma}$ е азбука и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. От корпуса $\mathcal{C}$ можем да научим множество различни изгладени n-грамни езикови модели, избирайки по различен начин константите $d_{\alpha a}$ и e_α. Сега ще разгледаме няколко техники за изглаждане на n-грамни езикови модели. Обзор и подробно сравнение на тези и много други изглаждащи техники могат да бъдат намерени в Chen и Goodman 1998.

1.5.1 Равномерно разпределение

Да разгледаме вероятностната мярка, разпределяща вероятностната маса равномерно между всичките параметри, тоест

$$P_U^{(n)}(a \mid \alpha) = \frac{1}{|\widetilde{\Sigma} \cup \{\$\}|} \text{ за } a \in \widetilde{\Sigma} \cup \{\$\} \text{ и } \alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}.$$

Индуктивно можем да проверим, че така дефинираните параметри са параметри на изгладен n-грамен езиков модел за $\mathcal{C}$.

- За $n = 1$, избирайки $d_a = \frac{1}{|\widetilde{\Sigma} \cup \{\$\}|}$ за $a \in \widetilde{\Sigma} \cup \{\$\}$, получаваме, че за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$

$$P_U^{(1)}(a \mid \varepsilon) = d_a.$$

- За $n \geq 2$, избирайки $d_{\alpha a} = \frac{1}{|\widetilde{\Sigma} \cup \{\$\}|}$ и $e_\alpha = 1$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, получаваме, че за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$

$$P_U^{(n)}(a \mid \alpha) = \begin{cases} P_U^{(|\beta|+1)}(a \mid \beta), & \text{ако } |\beta| < n-1 \\ d_{\alpha a}, & \text{ако } |\beta| = n-1 \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P_U^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases}$$

където $\beta = g_{\mathcal{C}}(\alpha)$, $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$.

1.5.2 Изглаждаща техника на Laplace

Една проста изглаждаща техника, която можем да приложим, е тази на Laplace. Изгладеният n-грамен езиков модел $P_L^{(n)}$, който получаваме, прилагайки тази техника за корпуса $\mathcal{C}$, има параметри, дефинирани за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ така

$$P_L^{(n)}(a \mid \alpha) = \frac{\#_{\mathcal{C}}(\alpha a) + 1}{\#_{\mathcal{C}}(\alpha\bullet) + |\widetilde{\Sigma} \cup \{\$\}|}.$$

Отново индуктивно можем да проверим, че параметрите $P_L^{(n)}(a \mid \alpha)$ са параметри на изгладен n-грамен езиков модел за $\mathcal{C}$.

- За $n = 1$, избирайки $d_a = \frac{\#_{\mathcal{C}}(a)+1}{\#_{\mathcal{C}}(\bullet)+|\widetilde{\Sigma}\cup\{\$\}|}$ за $a \in \widetilde{\Sigma} \cup \{\$\}$, получаваме, че за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$
$$P_L^{(1)}(a \mid \varepsilon) = d_a.$$
- За $n \geq 2$, избирайки $d_{\alpha a} = \frac{\#_{\mathcal{C}}(\alpha a)+1}{\#_{\mathcal{C}}(\alpha\bullet)+|\widetilde{\Sigma}\cup\{\$\}|}$ и $e_\alpha = \frac{|\widetilde{\Sigma}\cup\{\$\}|}{\#_{\mathcal{C}}(\alpha\bullet)+|\widetilde{\Sigma}\cup\{\$\}|}$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\}\circ\widetilde{\Sigma}^\star\}$, получаваме, че за всяко $a \in \widetilde{\Sigma}\cup\{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\}\circ\widetilde{\Sigma}^\star\}$
$$P_L^{(n)}(a \mid \alpha) = \begin{cases} P_U^{(|\beta|+1)}(a \mid \beta), & \text{ако } |\beta| < n-1 \\ d_{\alpha a}, & \text{ако } |\beta| = n-1 \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P_U^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases}$$
където $\beta = g_{\mathcal{C}}(\alpha)$, $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$.

1.5.3 Изглаждаща техника на Witten-Bell

Като пример за по-сложна изглаждаща техника да разгледаме изглаждащата техника на Witten-Bell. Изгладеният n-грамен езиков модел $P_{WB}^{(n)}$, който получаваме, прилагайки тази техника за корпуса $\mathcal{C}$, има параметри, дефинирани за $a \in \widetilde{\Sigma}\cup\{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\}\circ\widetilde{\Sigma}^\star\}$ така
$$P_{WB}^{(n)}(a \mid \alpha) = \frac{\#_{\mathcal{C}}(g_{\mathcal{C}}(\alpha)a)}{\#_{\mathcal{C}}(g_{\mathcal{C}}(\alpha)\bullet) + N_{+1}(g_{\mathcal{C}}(\alpha)\bullet)} + \frac{N_{+1}(g_{\mathcal{C}}(\alpha)\bullet)}{\#_{\mathcal{C}}(g_{\mathcal{C}}(\alpha)\bullet) + N_{+1}(g_{\mathcal{C}}(\alpha)\bullet)} X,$$
където
$$X = \begin{cases} P_{WB}^{(|\alpha'|+1)}(a \mid \alpha'), & \text{ако } |g_{\mathcal{C}}(\alpha)| > 0 \\ \frac{1}{|\widetilde{\Sigma}\cup\{\$\}|}, & \text{иначе} \end{cases}$$
и $\alpha' \succeq g_{\mathcal{C}}(\alpha)$, $|\alpha'| = |g_{\mathcal{C}}(\alpha)| - 1$. Изразът $N_{+1}(\alpha\bullet)$ представлява броя на уникалните продължения на историята α в корпуса $\mathcal{C}$. Формално за $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\}\circ\widetilde{\Sigma}^\star\}$ дефинираме
$$N_{+1}(\alpha\bullet) = |\{a \in \widetilde{\Sigma}\cup\{\$\} \mid \#_{\mathcal{C}}(\alpha a) > 0\}|.$$
Параметрите $P_{WB}^{(n)}(a \mid \alpha)$ също отговарят на условията от дефиницията на изгладен n-грамен езиков модел за $\mathcal{C}$.

- За $n = 1$, избирайки $d_a = \frac{\#_{\mathcal{C}}(a)}{\#_{\mathcal{C}}(\bullet)+N_{+1}(\bullet)} + \frac{N_{+1}(\bullet)}{\#_{\mathcal{C}}(\bullet)+N_{+1}(\bullet)}\frac{1}{|\widetilde{\Sigma}\cup\{\$\}|}$ за $a \in \widetilde{\Sigma}\cup\{\$\}$, получаваме, че за всяко $a \in \widetilde{\Sigma}\cup\{\$\}$
$$P_{WB}^{(1)}(a \mid \varepsilon) = d_a.$$

- За $n \geq 2$, избирайки $d_{\alpha a} = \frac{\#_{\mathcal{C}}(\alpha a)}{\#_{\mathcal{C}}(\alpha\bullet)+N_{+1}(\alpha\bullet)} + \frac{N_{+1}(\alpha\bullet)}{\#_{\mathcal{C}}(\alpha\bullet)+N_{+1}(\alpha\bullet)} P_{WB}^{(n-1)}(a \mid \alpha')$ и $e_\alpha = \frac{N_{+1}(\alpha\bullet)}{\#_{\mathcal{C}}(\alpha\bullet)+N_{+1}(\alpha\bullet)}$ за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, получаваме, че за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$

$$P_{WB}^{(n)}(a \mid \alpha) = \begin{cases} P_{WB}^{(|\beta|+1)}(a \mid \beta), & \text{ако } |\beta| < n-1 \\ d_{\alpha a}, & \text{ако } |\beta| = n-1 \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P_{WB}^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases}$$

където $|\beta| = g_{\mathcal{C}}(\alpha)$, $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$.

1.6 Изходно представяне на изгладени n-грамни езикови модели

ARPA е текстов формат за описание на изгладени n-грамни езикови модели (в смисъла, в който ние ги дефинирахме), създаден от Doug Paul. Повечето инструменти за научаване на езикови модели от корпус поддържат този формат и затова основната полза от него е оперативната съвместимост, която предоставя.

Нека да разгледаме едно описание в ARPA формат на изгладен n-грамен езиков модел $P^{(n)}$ за корпус $\mathcal{C}$ над речника $\widetilde{\Sigma}$. То се състои от две части – заглавна част и тяло. Заглавната част съдържа информация за стойностите $c_1, c_2, \ldots, c_n$, където

- $c_1 = |\widetilde{\Sigma} \cup \{\hat{}, \$\}|$;
- $c_i = |\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^i \mid \#_{\mathcal{C}}(\alpha) > 0\}|$ за $2 \leq i \leq n$.

Тялото се състои от n части. i-тата част за $1 \leq i \leq n$ представлява списък от наредени тройки с дължина c_i. Всяка от тези тройки съответства на някое $\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^i$ (такова че α се среща в корпуса $\mathcal{C}$, в случай че $i > 1$) и има вида $(d_\alpha, \alpha, e_\alpha)$, където d_α и e_α са константите от дефиницията на изгладен n-грамен езиков модел. [5] Разбира се, когато α не съответства на история, която се среща в корпуса, тоест $\alpha \notin \bigcup_{i=2}^{n} \{h_i(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$, стойността e_α е без значение. [6] Можем да забележим, че всичките наредени тройки от i-тата част на тялото представят точно $D_i \setminus D_{i-1}$ и $E_{i+1} \setminus E_i$, където D_k, E_k за $k \in \mathbb{N}^+$ са функциите от твърдение 1.6 и считаме, че $D_0 = E_{n+1} = \varnothing$. [7] Така можем да заключим, че информацията, която кодира ARPA описанието на $P^{(n)}$, представя точно функциите $D_n = \bigcup_{i=1}^{n} D_i \setminus D_{i-1}$ и $E_n = \bigcup_{i=1}^{n} E_{i+1} \setminus E_i$. Тоест от твърдение 1.6 следва, че едно описание в ARPA формат, еднозначно определя изгладен n-грамен езиков модел, ако параметрите, които то кодира, удовлетворяват необходимото и достатъчно условие, формулирано в твърдение 1.5.

В следващите глави ще разгледаме различни представяния на изгладени n-грамни езикови модели, като за всички тях ще представим псевдокод, описващ тяхното конструиране. В псевдокода ще считаме, че е даден изгладен n-грамен езиков модел в гореописания формат. По-конкретно, ще считаме, че изходното представяне на дадения изгладен n-грамен езиков модел е масив P с размер n. За $1 \leq i \leq n$ ще считаме, че $P[i]$ е свързан списък, съдържащ

5. Стандартно вместо d_α и e_α във формата се използват $\log_{10} d_\alpha$ и $\log_{10} e_\alpha$. Тъй като в случая този детайл е несъществен, ще го пренебрегнем.

6. Това са α, за които $\alpha \in \widetilde{\Sigma} \cup \{\hat{}, \$\}$ и $\#_{\mathcal{C}}(\alpha) = 0$ или $\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^i$ за $1 \leq i \leq n$ и $\$ \succeq \alpha$. По принцип за такива α в ARPA формата отсъства стойността e_α.

7. Изключение прави единствено думата $\hat{}$, за която има наредена тройка в ARPA описанието, а съотвено и стойност $d_{\hat{}}$, но която не е представена от функцията D_1. Разбира се, трябва да пренебрегнем и стойностите e_α за $\alpha \notin \bigcup_{i=2}^{n} \{h_i(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$.

всички наредени тройки от i-тата част на тялото на ARPA описанието на езиковия модел. Дължината на P ще означаваме с $|P|$, а дължината на всеки свързан списък $P[i]$ за $1 \leq i \leq n$ ще означаваме с $|P[i]|$. За всяка тройка $(d_\alpha, \alpha, e_\alpha)$ от списъка $P[i]$ за $1 \leq i \leq n$ ще предполагаме, че изречението α е представено като масив с дължина i, където $\alpha[j]$ за $1 \leq j \leq i$ е естествено число, кодиращо j-тата дума на изречението α. [8]

8. Реално в ARPA формата всяка дума a се представя чрез последователността си от символи. Ние ще предполагаме, че предварително всички думи от $\widetilde{\Sigma} \cup \{\hat{}, \$\}$ са били прекодирани в естествени числа.

2 Представяне на езикови модели чрез преобразуватели

Езиковите модели се използват в много задачи свързани с обработка на естествен език, като например разпознаване на реч, машинен превод, оптично разпознаване на символи и извличане на информация. Поради тази причина задачата за ефективно представяне на езикови модели е значима. Под ефективно представяне имаме предвид представяне, използващо „малко" ресурси – в случая време (за намиране на вероятността, приписана на дадено изречение) и памет.

Представянията, които ние ще разгледаме, са автоматни. От класическата теория на крайните автомати (Hopcroft и Ullman 1979) знаем, че всеки краен автомат може да бъде детерминиран. Детерминираните крайни автомати могат да бъдат траверсирани ефективно – за линейно време относно дължината на входа. Също така знаем, че всеки детерминиран краен автомат може да бъде минимизиран, тоест да бъде преобразуван в детерминиран краен автомат с минимален брой състояния измежду всички детерминирани крайни автомати, еквивалентни на изходния. Изхождайки от тези благоприятни свойства на крайните автомати, бихме искали да строим крайни детерминирани машини, които да представят езикови модели и които да можем в някакъв смисъл да „минимизираме".

2.1 Преобразуватели

Класическите крайни автомати представят функции от $\Sigma^\star$ в $\{0, 1\}$, където Σ е азбука. В случая искаме машина, която да може да представя езиков модел, тоест фунцкия от $\Sigma^\star$ в $[0, 1]$. За целта ще използваме разширение на понятието „детерминиран краен автомат", което ще наричаме „преобразувател". [1] Допълнителното, което преобразувателите имат спрямо детерминираните крайни автомати, е начален изход и изходи по преходите, тоест всеки преход освен вход – буква от входната азбука, с която се следва този преход, има и изход. Началният изход и изходите по преходите на даден преобразувател ще изискваме да бъдат елементи на моноид.

Определение 2.1. *Моноид* наричаме тройка $(M, \cdot_M, e_M)$, където

- $M \neq \varnothing$ е носителят на моноида;

1. В литературата (Mihov и Schulz, 2018 (in press)) терминът „преобразувател" има по-общо значение от това, което ние ще му припишем. По-конкретно – не се изисква да бъде детерминиран и да има начален изход. Това, което ние ще наричаме „преобразувател", отговаря на понятието „детерминиран преобразувател" с начален изход в Mihov и Schulz, 2018 (in press).

- $\cdot_M \colon M \to M$ е тотална асоциативна функция, тоест за всеки $a, b, c \in M$

$$(a \cdot_M b) \cdot_M c = a \cdot_M (b \cdot_M c);$$

- $e_M \in M$ е единичен елемент, тоест за всяко $a \in M$

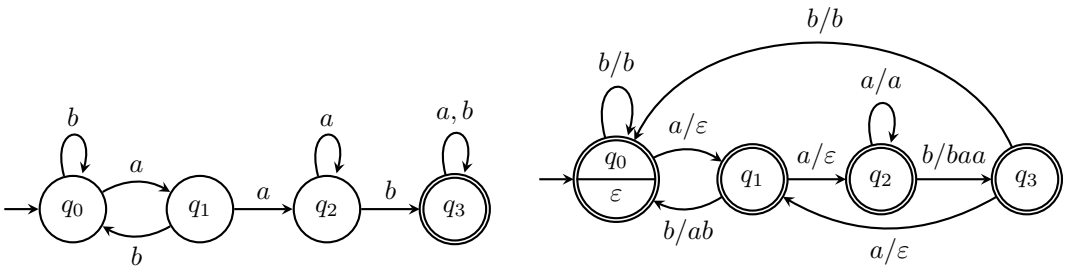
$$a \cdot_M e_M = e_M \cdot_M a = a.$$

Ще използваме ръкописни главни латински букви (като $\mathcal{M}$), за да означаваме моноиди. Носителя на моноид $\mathcal{M}$ ще означаваме със същата главна латинска буква, която обаче не е ръкописна (в случая M). За да опростим записа, ще си позволяваме да пропускаме долните индекси в записите $\cdot_M$ и e_M, когато моноидът се подразбира от контекста.

Определение 2.2. *Преобразувател* наричаме осморка $(\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$, където

- Σ е входна азбука;
- $\mathcal{M}$ е моноид;
- Q е крайно множество от състояния;
- $s \in Q$ е начално състояние;
- $F \subseteq Q$ е множество от финални състояния;
- $\delta \colon Q \times \Sigma \to Q$ е частична функция на преходите;
- $\lambda \colon Q \times \Sigma \to M$ е частична функция на изходите;
- $\iota \in M$ е начален изход

и $\mathrm{Dom}(\delta) = \mathrm{Dom}(\lambda)$.



Фигура 2.1: Примери за (а) детерминиран краен автомат $\mathcal{A}$ и (б) преобразувател $\mathcal{T}$ с изходи от моноида $(\{a, b\}^\star, \cdot, \varepsilon)$, където $\cdot$ е операцията „конкатенация на думи".

Преобразувателите ще изобразяваме с диаграми подобни на тези, с които се изобразяват крайните автомати. На фигура 2.1 са представени диаграми на детерминиран краен автомат и на преобразувател с изходи от моноида $(\{a, b\}^\star, \cdot, \varepsilon)$, където $\cdot$ е операцията „конкатенация на думи". Състоянията, началното състояние, финалните състояния и преходите (с изключение на изходите) на преобразувателите се изобразяват по същия начин като при крайните автомати. Началният изход се записва под името на началното състояние (на фигура 2.1(б)

началното състояние е q_0, а началният изход е ε). Входната буква и изходът на даден преход се записват върху съответстващата му стрелка и се разделят с /, като входната буква стои вляво, а изходът вдясно.

Траверсирането на преобразувател с дадена входна дума се осъществява по начина, по който се траверсира краен автомат. Преобразувателят стартира в началното си състояние и чете входната дума буква по буква. При прочитането на следващата буква σ преобразувателят преминава (прави преход) от текущото си състояние q в състоянието, определено от q, σ и δ, по-точно – в $\delta(q, \sigma)$. Стандартно ще дефинираме функцията $\delta^\star$ – разширението на δ върху $Q \times \Sigma^\star$, която ще отразява смяната на текущото състояние на преобразувателя при прочитане на последователност от входни букви.

Определение 2.3. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Фунцкията $\delta^\star \colon Q \times \Sigma^\star \to Q$ дефинираме индуктивно

- $\delta^\star(q, \varepsilon) = q$ за всяко $q \in Q$;
- $\delta^\star(q, \alpha a) = \delta(\delta^\star(q, \alpha), a)$ за всяко $q \in Q$, $\alpha \in \Sigma^\star$ и $a \in \Sigma$.

Резултатът от едно траверсиране на детерминиран краен автомат с дадена входна дума е отговор „да/не" в зависимост от това дали траверсирането е завършило преждевременно (поради недефиниран преход) и ако не е, от това дали състоянието, в което е завършило траверсирането, е финално или не. При преобразувателите резултатът от едно траверсиране

- или е недефиниран, ако траверсирането завърши преждевременно или завърши в нефинално състояние;
- или представлява акумулираните с операцията на моноида изходи на направените при траверсирането преходи (заедно с началния изход), ако траверсирането завърши във финално състояние.

Ще въведем функцията $\lambda^\star \colon Q \times \Sigma^\star \to M$, която да отразява акумулирането на изходите на направените при прочитане на дадена последователност от входни букви преходи (без да взима предвид началния изход), и функцията $O_{\mathcal{T}} \colon \Sigma^\star \to M$, която да отразява резултата от траверсиране на преобразувателя $\mathcal{T}$ с дадена входна дума.

Определение 2.4. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Функцията $\lambda^\star \colon Q \times \Sigma^\star \to M$ дефинираме индуктивно

- $\lambda^\star(q, \varepsilon) = e$ за всяко $q \in Q$;
- $\lambda^\star(q, \alpha a) = \lambda^\star(q, \alpha) \cdot \lambda(\delta^\star(q, \alpha), a)$ за всяко $q \in Q$, $\alpha \in \Sigma^\star$ и $a \in \Sigma$.

Определение 2.5. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Функцията $O_{\mathcal{T}} \colon \Sigma^\star \to M$, дефинирана за $\alpha \in \Sigma^\star$ така

$$O_{\mathcal{T}}(\alpha) = \begin{cases} \iota \cdot \lambda^\star(s, \alpha), & \text{ако } \delta^\star(s, \alpha) \in F \\ \neg\, !, & \text{иначе} \end{cases}$$

наричаме *функцията, представена от преобразувателя* $\mathcal{T}$.

На примерите от фигура 2.1 детерминираният краен автомат $\mathcal{A}$ разпознава всички думи над $\{a, b\}$, които съдържат aab като поддума, а преобразувателят $\mathcal{T}$ представя функцията $O_{\mathcal{T}} \colon \{a, b\}^\star \to \{a, b\}^\star$, съпоставяща на всяка дума $\alpha \in \{a, b\}^\star$ думата, получена от α чрез заместване на всяко срещане на aab с baa.

Можем да забележим, че началните изходи на преобразувателите не увеличават изразителността им, ако разглеждаме само функции с домейн несъдържащ ε. Това е така, тъй като винаги можем да добавим ново начално състояние, което има единичен начален изход и същите преходи като старото начално състояние, с разликата, че към изходите им отляво е добавен старият начален изход. [2]

Твърдение 2.1. *Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Тогава преобразувателят $\mathcal{T}' = (\Sigma, \mathcal{M}, Q', s', F, \delta', \lambda', e)$, където*

- $Q' = Q \cup \{s'\}$;
- $s' \notin Q$;
- *за $q \in Q'$ и $\sigma \in \Sigma$*

$$\delta'(q,\sigma) = \begin{cases} \delta(q,\sigma), & \text{ако } q \in Q \\ \delta(s,\sigma), & \text{ако } q = s' \end{cases} \qquad \lambda'(q,\sigma) = \begin{cases} \lambda(q,\sigma), & \text{ако } q \in Q \\ \iota \cdot \lambda(s,\sigma), & \text{ако } q = s' \end{cases}$$

е такъв че

$$(\forall \alpha \in \Sigma^+)(O_{\mathcal{T}}(\alpha) = O_{\mathcal{T}'}(\alpha)).$$

Доказателство. С индукция по $|\alpha|$ ще докажем, че

$$(\forall \alpha \in \Sigma^+)(\delta'^\star(s',\alpha) = \delta^\star(s,\alpha) \wedge \lambda'^\star(s',\alpha) = \iota \cdot \lambda^\star(s,\alpha)).$$

$|\alpha| = 1$

Тогава $\alpha \in \Sigma$ и

1. $\delta'^\star(s',\alpha) = \delta'(s',\alpha) = \delta(s,\alpha) = \delta^\star(s,\alpha)$;
2. $\lambda'^\star(s',\alpha) = \lambda'(s',\alpha) = \iota \cdot \lambda(s,\alpha) = \iota \cdot \lambda^\star(s,\alpha)$.

$|\alpha| \rightsquigarrow |\alpha| + 1$

Нека $\alpha = \alpha' a$ и за всяка дума с дължина $|\alpha'|$ твърдението е в сила. Тогава

1. $\begin{aligned}\delta'^\star(s',\alpha) &= \delta'^\star(s',\alpha' a) = \delta'(\delta'^\star(s',\alpha'),a) = \delta'(\delta^\star(s,\alpha'),a) \\ &= \delta(\delta^\star(s,\alpha'),a) = \delta^\star(s,\alpha' a) = \delta^\star(s,\alpha);\end{aligned}$
2. $\begin{aligned}\lambda'^\star(s',\alpha) &= \lambda'^\star(s',\alpha' a) = \lambda'^\star(s',\alpha') \cdot \lambda'(\delta'^\star(s',\alpha'),a) \\ &= \iota \cdot \lambda^\star(s,\alpha') \cdot \lambda'(\delta^\star(s,\alpha'),a) = \iota \cdot \lambda^\star(s,\alpha') \cdot \lambda(\delta^\star(s,\alpha'),a) \\ &= \iota \cdot \lambda^\star(s,\alpha' a) = \iota \cdot \lambda^\star(s,\alpha).\end{aligned}$

Сега за всяка дума $\alpha \in \Sigma^+$ е в сила, че

$$\alpha \in \mathrm{Dom}(O_{\mathcal{T}}) \iff \delta^\star(s,\alpha) \in F \iff \delta'^\star(s',\alpha) \in F \iff \alpha \in \mathrm{Dom}(O_{\mathcal{T}'})$$

и ако $\alpha \in \mathrm{Dom}(O_{\mathcal{T}})$, то

$$O_{\mathcal{T}}(\alpha) = \iota \cdot \lambda^\star(s,\alpha) = \lambda'^\star(s',\alpha) = e \cdot \lambda'^\star(s',\alpha) = O_{\mathcal{T}'}(\alpha). \qquad \square$$

Причината, поради която позволяваме преобразувателите да имат начални изходи, е улеснение при въвеждане на понятиятието „каноничен преобразувател".

2. Директно от дефиницията на преобразувател без начален изход (Mihov и Schulz, 2018 (in press)) следва, че за всеки преобразувател с единичен начален изход съществува еквивалентен с точност до ε преобразувател без начален изход. Два преобразувателя са еквивалентни с точност до ε, ако функциите, които те представят, съвпадат, с изключение евентуално на стойностите, които те приписват на празната дума.

2.2 Представяне на езикови модели чрез преобразователи на ниво думи

Чрез директна проверка на условията от дефиницията за моноид, се вижда, че $([0,1],\cdot,1)$, където $\cdot$ е операцията „умножение на реални числа", е моноид. Това е моноидът, който ще използваме, когато представяме изгладени n-грамни езикови модели с преобразуватели, и ще означаваме с $\mathcal{R}_{[0,1]}$.

Сега ще се убедим, че всеки изгладен n-грамен езиков модел може да бъде представен чрез преобразувател на ниво думи, тоест преобразувател, използващ речника на езиковия модел за входна азбука. [3] За целта първо ще въведем запис, който ще представя инфикс на произведението

$$\prod_{i=2}^{m+2} P^{(n)}(a_i \mid h_n(a_1 a_2 \dots a_{i-1})),$$

представящо вероятността на изречение $\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \dots a_{m+2}$, приписана му от изгладен n-грамен езиков модел $P^{(n)}$. Дефиницията на записа ще използва само параметри с истории, които са се срещнали в корпуса.

Определение 2.6. Нека $\widetilde{\Sigma}$ е азбука и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. Нека $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава за $\beta \in \mathrm{Pref}(\widetilde{\Sigma}^\star \circ \{\$\})$, $\beta = b_1 b_2 \dots b_m$ и $\alpha \in \bigcup_{i=1}^{n} \{h_i(\gamma) \mid \gamma \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ дефинираме

$$P^{(n)}(\beta \mid \alpha) = \prod_{i=1}^{m} P^{(|\gamma_i|+1)}(b_i \mid \gamma_i),$$

където $\gamma_i = g_{\mathcal{C}}(h_n(\alpha b_1 b_2 \dots b_{i-1}))$ за $1 \leq i \leq m$.

Можем да се убедим, че всеки множител $P^{(|\gamma_i|+1)}(b_i \mid \gamma_i)$ от произведението в горната дефиниция е параметър, тъй като за всяко $i \in \{1, 2, \dots, m\}$ е в сила, че

1. $b_i \in \widetilde{\Sigma} \cup \{\$\}$;
2. $\gamma b_1 b_2 \dots b_{i-1} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star$;
3. $\gamma_i = h_{|\gamma_i|+1}(\alpha b_1 b_2 \dots b_{i-1}) = h_{|\gamma_i|+1}(\gamma b_1 b_2 \dots b_{i-1})$,

където $\gamma \in \{\hat{}\} \circ \widetilde{\Sigma}^\star$ е това, за което $\alpha = h_{|\alpha|+1}(\gamma)$.

Многозначност между записа на параметъра $P^{(n)}(b \mid \alpha)$ и нововъведения запис $P^{(n)}(\beta \mid \alpha)$ за $\beta = b$ не възниква, тъй като те носят една и съща стойност. Използвайки новия запис, можем да изразим вероятността на произволно изречение от вероятностното пространство по следния начин.

Твърдение 2.2. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава*

$$(\forall \alpha \in \widetilde{\Sigma}^\star)(P^{(n)}(\{\hat{}\alpha\$\}) = P^{(n)}(\alpha\$ \mid h_n(\hat{}))).$$

Доказателство. Нека $\alpha \in \widetilde{\Sigma}^\star$ и $\alpha = a_1 a_2 \dots a_m$. Тогава

$$P^{(n)}(\{\hat{}\alpha\$\}) = \left(\prod_{i=1}^{m} P^{(n)}(a_i \mid h_n(\hat{}a_1 \dots a_{i-1})) \right) P^{(n)}(\$ \mid h_n(\hat{}\alpha))$$

3. Всъщност е вярно нещо повече – всеки n-грамен езиков модел може да бъде представен чрез преобразувател на ниво думи. Ние ще покажем твърдението само за изгладени n-грамни езикови модели, тъй като, както отбелязахме в част 1.4, това са n-грамните езикови модели, които се използват на практика.

$$P^{(n)}(\alpha\$ \mid h_n(\hat{}\,)) = \left(\prod_{i=1}^{m} P^{(|\gamma_i|+1)}(a_i \mid \gamma_i)\right) P^{(|\gamma_{m+1}|+1)}(\$ \mid \gamma_{m+1}),$$

където $\gamma_i = g_{\mathcal{C}}(h_n(\hat{}\,a_1a_2\ldots a_{i-1}))$ за $1 \leq i \leq m+1$. [4]

За да покажем, че $P^{(n)}(\{\hat{}\,\alpha\$\}) = P^{(n)}(\alpha\$ \mid h_n(\hat{}\,))$, е достатъчно да видим, че

$$(\forall a \in \widetilde{\Sigma} \cup \{\$\})(\forall \alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\,\} \circ \widetilde{\Sigma}^\star\})(P^{(n)}(a \mid \alpha) = P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha))).$$

$n = 1$

$$P^{(1)}(a \mid \alpha) = P^{(1)}(a \mid \varepsilon) = P^{(|g_{\mathcal{C}}(\varepsilon)|+1)}(a \mid g_{\mathcal{C}}(\varepsilon)) = P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)).$$

$n > 1$

$$\begin{aligned}
P^{(n)}(a \mid \alpha) &= \begin{cases} P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)), & \text{ако } |g_{\mathcal{C}}(\alpha)| < n-1 \\ d_{\alpha a}, & \text{ако } |g_{\mathcal{C}}(\alpha)| = n-1 \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases} \\
&= \begin{cases} P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)), & \text{ако } |g_{\mathcal{C}}(\alpha)| < n-1 \\ P^{(n)}(a \mid \alpha), & \text{ако } |g_{\mathcal{C}}(\alpha)| = n-1 \end{cases} \\
&= \begin{cases} P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)), & \text{ако } |g_{\mathcal{C}}(\alpha)| < n-1 \\ P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)), & \text{ако } |g_{\mathcal{C}}(\alpha)| = n-1 \end{cases} \\
&= P^{(|g_{\mathcal{C}}(\alpha)|+1)}(a \mid g_{\mathcal{C}}(\alpha)).
\end{aligned}$$

Тук $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$ (такъв суфикс съществува, тъй като $n > 1 \implies |\alpha| > 0$). Предпоследното равенство следва от това, че щом $|\alpha| < n$ и $g_{\mathcal{C}}(\alpha) = n-1$, то $|\alpha| = n-1$ и $g_{\mathcal{C}}(\alpha) = \alpha$. □

Нововъведеният запис отразява факта, че се нуждаем единствено от параметрите, чиито истории са се срещнали в корпуса. Използвайки това знание, ще построим преобразувател със състояния – всички истории, срещнати в корпуса, и преходи, отразяващи преминаването (при прочитане на дума a) от дадена история α в най-дългата история суфикс на αa, която се е срещнала в корпуса.

Определение 2.7. Нека $\widetilde{\Sigma}$ е азбука и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. Нека $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава преобразувателя

$n = 1$

$(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, \{\varepsilon, \$\}, \varepsilon, \{\$\}, \delta, \lambda, 1)$, където за $\alpha \in Q$ и $a \in \widetilde{\Sigma} \cup \{\$\}$

$$\delta(\alpha, a) = \begin{cases} \varepsilon, & \text{ако } \alpha a \in \widetilde{\Sigma} \\ \$, & \text{ако } \alpha a = \$ \\ \neg\,!, & \text{иначе} \end{cases} \qquad \lambda(\alpha, a) = \begin{cases} P^{(1)}(a \mid \alpha), & \text{ако } !\,\delta(\alpha, a) \\ \neg\,!, & \text{иначе} \end{cases}$$

$n \geq 2$

$(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, \hat{}\,, F, \delta, \lambda, 1)$, където

- $Q = \{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}\,, \$\})^\star \mid |\alpha| < n \wedge \#_{\mathcal{C}}(\alpha) > 0\}$;

4. Лесно се вижда, че $h_n(h_n(\hat{}\,)a_1a_2\ldots a_{i-1}) = h_n(\hat{}\,a_1a_2\ldots a_{i-1})$ за всяко $n \in \mathbb{N}^+$ и всяко $1 \leq i \leq m+1$.

- $F = \{\alpha \in Q \mid \$ \succeq \alpha\}$;
- за $\alpha \in Q$ и $a \in \widetilde{\Sigma} \cup \{\$\}$

$$\delta(\alpha, a) = \begin{cases} g_{\mathcal{C}}(h_n(\alpha a)), & \text{ако } \alpha \notin F \\ \neg\,!, & \text{иначе} \end{cases} \qquad \lambda(\alpha, a) = \begin{cases} P^{(|\alpha|+1)}(a \mid \alpha), & \text{ако } !\,\delta(\alpha, a) \\ \neg\,!, & \text{иначе} \end{cases}$$

наричаме *преобразувателят на ниво думи за* $P^{(n)}$.

Сега ще покажем, че преобразувателите от определение 2.7 наистина представят съответстващите им изгладени n-грамни езикови модели. Първо ще опишем поведението на преобразувателите при траверсиране с дадено входно изречение.

Твърдение 2.3. *Нека* $\mathcal{C}$ *е корпус над азбуката* $\widetilde{\Sigma}$ *и* $P^{(1)}$ *е изгладен* 1*-грамен езиков модел за* $\mathcal{C}$*. Нека* $(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, \{\varepsilon, \$\}, \varepsilon, \{\$\}, \delta, \lambda, 1)$ *е преобразувателят на ниво думи за* $P^{(1)}$*. Тогава за всяко* $\alpha \in \mathrm{Pref}(\widetilde{\Sigma}^\star \circ \{\$\})$ *е в сила, че*

$$(\delta^\star(\varepsilon, \alpha) = \$ \iff \$ \succeq \alpha) \wedge \lambda^\star(\varepsilon, \alpha) = P^{(1)}(\alpha \mid \varepsilon).$$

Доказателство. Твърдението ще докажем с индукция по $|\alpha|$.

$|\alpha| = 0$
Тогава $\alpha = \varepsilon$ и

1. $\delta^\star(\varepsilon, \alpha) = \delta^\star(\varepsilon, \varepsilon) = \varepsilon$ и $\$ \not\succeq \varepsilon$;
2. $\lambda^\star(\varepsilon, \alpha) = \lambda^\star(\varepsilon, \varepsilon) = 1 = P^{(1)}(\varepsilon \mid \varepsilon)$.

$|\alpha| \rightsquigarrow |\alpha| + 1$
Нека $\alpha = \alpha' a$ и за всяка дума с дължина $|\alpha'|$ твърдението е в сила. Щом $\alpha = \alpha' a$ и $\alpha \in \mathrm{Pref}(\widetilde{\Sigma}^\star \circ \{\$\})$, то $\alpha' \in \widetilde{\Sigma}^\star$ и $\$ \not\succeq \alpha'$. Тогава

1. $\delta^\star(\varepsilon, \alpha) = \delta^\star(\varepsilon, \alpha' a) = \delta(\delta^\star(\varepsilon, \alpha'), a) = \delta(\varepsilon, a) = \$ \iff a = \$ \iff \$ \succeq \alpha$;
2. $\begin{aligned} \lambda^\star(\varepsilon, \alpha) &= \lambda^\star(\varepsilon, \alpha' a) = \lambda^\star(\varepsilon, \alpha') \cdot \lambda(\delta^\star(\varepsilon, \alpha'), a) = P^{(1)}(\alpha' \mid \varepsilon) \cdot \lambda(\varepsilon, a) \\ &= P^{(1)}(\alpha' \mid \varepsilon) \cdot P^{(1)}(a \mid \varepsilon) = P^{(1)}(\alpha' \mid \varepsilon) \cdot P^{(|g_{\mathcal{C}}(\varepsilon)|+1)}(a \mid g_{\mathcal{C}}(\varepsilon)) \\ &= P^{(1)}(\alpha' \mid \varepsilon) \cdot P^{(|g_{\mathcal{C}}(h_1(\alpha'))|+1)}(a \mid g_{\mathcal{C}}(h_1(\alpha'))) = P^{(1)}(\alpha' a \mid \varepsilon) \\ &= P^{(1)}(\alpha \mid \varepsilon). \end{aligned}$ □

Сега ще разгледаме едно свойство на функциите $g_{\mathcal{C}}$ и h_n, което ще използваме в доказателството на твърдението, описващо поведението на преобразувателя на ниво думи за изгладен n-грамен езиков модел ($n \geq 2$) при траверсиране с дадено входно изречение.

Твърдение 2.4. *Нека* $n \in \mathbb{N}^+$ *и* $\mathcal{C}$ *е корпус над азбуката* $\widetilde{\Sigma}$*. Тогава*

$$(\forall \alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star)(\forall a \in \widetilde{\Sigma} \cup \{\hat{}, \$\})(g_{\mathcal{C}}(h_n(g_{\mathcal{C}}(h_n(\alpha))a)) = g_{\mathcal{C}}(h_n(\alpha a))).$$

Доказателство. Нека $\alpha \in (\widetilde{\Sigma} \cup \{, \$\})^\star$, $\alpha = a_1 a_2 \ldots a_m$ и $a \in \widetilde{\Sigma} \cup \{\hat{}, \$\}$. Нека $1 \leq i \leq m+1$ е такова, че $g_{\mathcal{C}}(\alpha) = a_i a_{i+1} \ldots a_m$.

1сл. $\boldsymbol{m - n + 2 < i}$

Тогава $g_{\mathcal{C}}(h_n(\alpha)) = g_{\mathcal{C}}(a_{\max\{1,m-n+2\}} \dots a_m) = a_i \dots a_m$, $|a_i \dots a_m a| = m - i + 2 < n$ и

$$\begin{aligned} g_{\mathcal{C}}(h_n(g_{\mathcal{C}}(h_n(\alpha))a)) &= g_{\mathcal{C}}(h_n(a_i \dots a_m a)) \\ &= g_{\mathcal{C}}(a_i \dots a_m a) \\ &= g_{\mathcal{C}}(a_{\max\{1,m-n+3\}} \dots a_m a) \\ &= g_{\mathcal{C}}(h_n(\alpha a)). \end{aligned}$$

Нека $j = \max\{1, m-n+3\}$. Тогава $j \leq i$ и $a_i \dots a_m a \succeq a_j \dots a_m a \succeq a_{\max\{1,m-n+2\}} \dots a_m$. Сега ще обясним защо предпоследното равенство е вярно, тоест защо $g_{\mathcal{C}}(a_i \dots a_m a) = g_{\mathcal{C}}(a_j \dots a_m a)$. Първо, ясно е, че ако $|g_{\mathcal{C}}(a_j \dots a_m a)| \leq |a_i \dots a_m a|$, то $g_{\mathcal{C}}(a_i \dots a_m a) = g_{\mathcal{C}}(a_j \dots a_m a)$, тъй като $g_{\mathcal{C}}(a_j \dots a_m a)$ е свидетел за суфикс на $a_i \dots a_m a$, който се е срещнал в корпуса, и в същото време $g_{\mathcal{C}}(a_i \dots a_m a)$ не може да е по-дълъг от този суфикс, тъй като това ще противоречи с масималността на $g_{\mathcal{C}}(a_j \dots a_m a)$. Второ, ако допуснем, че $|g_{\mathcal{C}}(a_j \dots a_m a)| > |a_i \dots a_m a|$, тоест $g_{\mathcal{C}}(a_j \dots a_m a) = a_k \dots a_m a$, където $j \leq k < i$, то стигаме до противоречие с факта, че $g_{\mathcal{C}}(a_{\max\{1,m-n+2\}} \dots a_m) = a_i \dots a_m$, защото $a_k \dots a_m$ е суфикс на $a_{\max\{1,m-n+2\}} \dots a_m$, който се е срещнал в корпуса и е по-дълъг от $a_i \dots a_m$.

2сл. $\boldsymbol{m - n + 2 \geq i}$

Тогава $g_{\mathcal{C}}(h_n(\alpha)) = g_{\mathcal{C}}(a_{\max\{1,m-n+2\}} \dots a_m) = a_{\max\{1,m-n+2\}} \dots a_m$ и

$$g_{\mathcal{C}}(h_n(g_{\mathcal{C}}(h_n(\alpha))a)) = g_{\mathcal{C}}(h_n(a_{\max\{1,m-n+2\}} \dots a_m a)) = g_{\mathcal{C}}(h_n(\alpha a)).$$

Последното равенство е вярно, тъй като ако $|\alpha| < n$, то $\max\{1, m - n + 2\} = 1$ и

$$h_n(a_{\max\{1,m-n+2\}} \dots a_m a) = h_n(a_1 \dots a_m a) = h_n(\alpha a).$$

В противен случай $\max\{1, m - n + 2\} = m - n + 2$ и

$$h_n(a_{\max\{1,m-n+2\}} \dots a_m a) = h_n(a_{m-n+2} \dots a_m a) = a_{m-n+3} \dots a_m a = h_n(\alpha a). \qquad \square$$

Твърдение 2.5. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$, където $n \geq 2$. Нека $(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, \hat{}, F, \delta, \lambda, 1)$ е преобразувателят на ниво думи за $P^{(n)}$. Тогава за всяко $\alpha \in \mathrm{Pref}(\widetilde{\Sigma}^\star \circ \{\$\})$ е в сила, че*

$$\delta^\star(\hat{}, \alpha) = g_{\mathcal{C}}(h_n(\hat{}\alpha)) \wedge \lambda^\star(\hat{}, \alpha) = P^{(n)}(\alpha \mid \hat{}).$$

Доказателство. Твърдението ще докажем с индукция по $|\alpha|$.

$\boldsymbol{|\alpha| = 0}$

Тогава $\alpha = \varepsilon$ и

1. $\delta^\star(\hat{}, \alpha) = \delta^\star(\hat{}, \varepsilon) = \hat{} = g_{\mathcal{C}}(\hat{}) = g_{\mathcal{C}}(h_n(\hat{})) = g_{\mathcal{C}}(h_n(\hat{}\alpha))$;
2. $\lambda^\star(\hat{}, \alpha) = \lambda^\star(\hat{}, \varepsilon) = 1 = P^{(n)}(\varepsilon \mid \hat{}) = P^{(n)}(\alpha \mid \hat{})$.

$\boldsymbol{|\alpha| \rightsquigarrow |\alpha| + 1}$

Нека $\alpha = \alpha' a$ и за всяка дума с дължина $|\alpha'|$ твърдението е в сила. Щом $\alpha = \alpha' a$ и $\alpha \in \mathrm{Pref}(\widetilde{\Sigma}^\star \circ \{\$\})$, то $\alpha' \in \widetilde{\Sigma}^\star$, $\$ \not\preceq \alpha'$ и $\$ \not\preceq \delta^\star(\hat{}, \alpha') = g_{\mathcal{C}}(h_n(\hat{}\alpha'))$. Тоест $\delta^\star(s, \alpha') \notin F$ и

1. $\begin{aligned} \delta^\star(\hat{}, \alpha) &= \delta^\star(\hat{}, \alpha' a) = \delta(\delta^\star(\hat{}, \alpha'), a) = \delta(g_{\mathcal{C}}(h_n(\hat{}\alpha')), a) \\ &= g_{\mathcal{C}}(h_n(g_{\mathcal{C}}(h_n(\hat{}\alpha'))a)) = g_{\mathcal{C}}(h_n(\hat{}\alpha' a)) = g_{\mathcal{C}}(h_n(\hat{}\alpha)); \end{aligned}$

2. $\lambda^\star(\hat{}, \alpha) = \lambda^\star(\hat{}, \alpha' a) = \lambda^\star(\hat{}, \alpha') \cdot \lambda(\delta^\star(\hat{}, \alpha'), a)$
$= P^{(n)}(\alpha' \mid \hat{}) \cdot \lambda(g_{\mathcal{C}}(h_n(\hat{}\alpha')), a)$
$= P^{(n)}(\alpha' \mid \hat{}) \cdot P^{(|g_{\mathcal{C}}(h_n(\hat{}\alpha'))|+1)}(a \mid g_{\mathcal{C}}(h_n(\hat{}\alpha')))$
$= P^{(n)}(\alpha' a \mid \hat{}) = P^{(n)}(\alpha \mid \hat{}).$ □

Сега, знаейки как се държат преобразувателите от определение 2.7 при траверсиране с дадено входно изречение, можем да заключим, че те представят съответстващите им изгладени n-грамни езикови модели.

Твърдение 2.6. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{T} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, 1)$ е преобразувателят на ниво думи за $P^{(n)}$. Тогава е в сила, че*

- $\mathrm{Dom}(O_{\mathcal{T}}) = \widetilde{\Sigma}^\star \circ \{\$\}$;
- $(\forall \alpha \in \widetilde{\Sigma}^\star)(O_{\mathcal{T}}(\alpha\$) = P^{(n)}(\{\hat{}\alpha\$\}))$.

Доказателство. Ако $\alpha \in \mathrm{Dom}(O_{\mathcal{T}})$, то $\delta^\star(s, \alpha) \in F$. Състоянията от F имат входящи преходи само с буква \$, следователно $\alpha = \alpha'\$$. Тъй като състоянията от F нямат изходящи преходи и са единствените състояния с входящ преход с буква \$, следва, че $\alpha' \in \widetilde{\Sigma}^\star$, тоест $\alpha \in \widetilde{\Sigma}^\star \circ \{\$\}$. Ако $\alpha \in \widetilde{\Sigma}^\star \circ \{\$\}$, то $\$ \succeq \alpha$ и съгласно твърдение 2.3 и твърдение 2.5, $\$ \succeq \delta^\star(s, \alpha)$ (при $n \geq 2$ очевидно $\$ \succeq h_n(\hat{}\alpha)$, а $\$ \succeq g_{\mathcal{C}}(h_n(\hat{}\alpha))$, тъй като $\mathcal{C}$ по дефиниция съдържа поне едно изречение и следователно $\#_{\mathcal{C}}(\$) > 0$). Тоест $\delta^\star(s, \alpha) \in F$ и $\alpha \in \mathrm{Dom}(O_{\mathcal{T}})$. Така показахме, че $\mathrm{Dom}(O_{\mathcal{T}}) = \widetilde{\Sigma}^\star \circ \{\$\}$.

За втората част от твърдението нека $\alpha \in \widetilde{\Sigma}^\star$. Тогава, позовавайки се на твърдение 2.3 и твърдение 2.5, заключаваме

$$O_{\mathcal{T}}(\alpha\$) = 1 \cdot \lambda^\star(s, \alpha\$) = \lambda^\star(h_n(\hat{}), \alpha\$) = P^{(n)}(\alpha\$ \mid h_n(\hat{})) = P^{(n)}(\{\hat{}\alpha\$\}).$$ □

На фигура 2.2 е изобразен преобразувателят на ниво думи за $P^{(2)}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\alpha\beta\alpha\$, \hat{}\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$. Вижда се как всяко състояние (срещната в корпуса история) има преход с всяка дума от речника, влизащ в състоянието, съответстващо на най-дългата история (след прочитане на думата), която има дължина ненадминаваща 1 и която се е срещнала в корпуса.

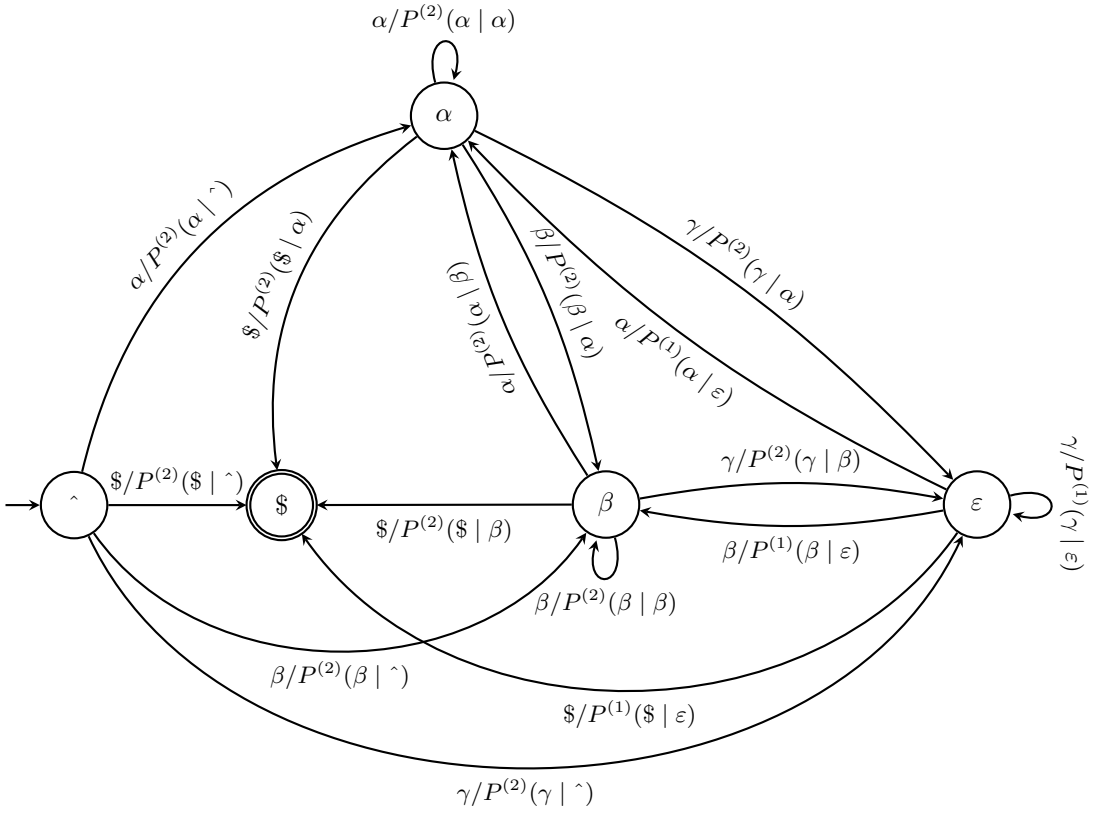
Дефинираните в определение 2.7 преобразуватели имат размер (брой състояния и преходи), зависещ мултилинейно от n, броя думи в корпуса и броя думи в речника.

Твърдение 2.7. *Нека $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава преобразувателят $(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, 1)$ на ниво думи за $P^{(n)}$ е такъв, че*

- $|Q| \in O(n \sum_{i=1}^m |\alpha_i|)$;
- $|\mathrm{Dom}(\delta)| \in O(n|\widetilde{\Sigma}| \sum_{i=1}^m |\alpha_i|)$.

Доказателство. При $n = 1$ състоянията на преобразувателя са 2, а изходящи преходи има само състоянието ε и то с всяка буква от $\widetilde{\Sigma} \cup \{\$\}$. Тоест, тъй като $\sum_{i=1}^m |\alpha_i| > 1$,

- $|Q| = 2 \in O(n \sum_{i=1}^m |\alpha_i|)$;
- $|\mathrm{Dom}(\delta)| = |\widetilde{\Sigma}| + 1 \in O(n|\widetilde{\Sigma}| \sum_{i=1}^m |\alpha_i|)$.



Фигура 2.2: Преобразувателят на ниво думи за $P^{(2)}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\alpha\beta\alpha\$$, $\hat{}\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$.

При $n \geq 2$ всяко състояние съответства на последователност от думи от $\widetilde{\Sigma} \cup \{\hat{}, \$\}$ с дължина по-малка от n, която се е срещнала в корпуса, и всяко нефинално състояние има изходящ преход с всяка буква от $\widetilde{\Sigma} \cup \{\$\}$.

Така за броя на състоянията заключаваме, че

$$\begin{aligned}
|Q| &= |\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |\alpha| < n \wedge \#_{\mathcal{C}}(\alpha) > 0\}| \\
&= |\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |\alpha| < n \wedge \sum_{i=1}^{m} \#_{\alpha_i}(\alpha) > 0\}| \\
&= |\bigcup_{i=1}^{m} \{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |\alpha| < n \wedge \#_{\alpha_i}(\alpha) > 0\}| \\
&= |\bigcup_{i=1}^{m} \bigcup_{j=0}^{n-1} \{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |\alpha| = j \wedge \#_{\alpha_i}(\alpha) > 0\}| \\
&\leq \sum_{i=1}^{m} \sum_{j=0}^{n-1} |\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |\alpha| = j \wedge \#_{\alpha_i}(\alpha) > 0\}|
\end{aligned}$$

$$\leq \sum_{i=1}^{m}\sum_{j=0}^{n-1}|\alpha_i|$$
$$= n\sum_{i=1}^{m}|\alpha_i| \in O(n\sum_{i=1}^{m}|\alpha_i|).$$

В последното неравенство използваме, че за $j \in \mathbb{N}$ и $1 \leq i \leq m$

$$|\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^0 \quad | \#_{\alpha_i}(\alpha)\}| = 1 \leq |\alpha_i|,$$
$$|\{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^{j+1} \mid \#_{\alpha_i}(\alpha)\}| = |\{a_k^i \dots a_{k+j}^i \mid 1 \leq k \leq |\alpha_i| - j\}| \leq |\alpha_i|,$$

където с a_k^i означаваме k-тия символ на α_i.

Сега за броя на преходите следва, че

$$|\operatorname{Dom}(\delta)| \leq |Q \times (\widetilde{\Sigma} \cup \{\$\})| = |Q||\widetilde{\Sigma} \cup \{\$\}| \in O(n|\widetilde{\Sigma}|\sum_{i=1}^{m}|\alpha_i|). \qquad \square$$

2.3 Базови структури от данни

Накрая на тази глава ще дадем псевдокод, описващ конструкцията от определение 2.7. В псевдокода, представен в тази и в останалите глави, ще използваме структурите от данни масив, свързан списък, матрица и хеш-таблица. [5] Тук ще опишем накратко операциите, които ще прилагаме върху тези структури от данни.

Нов масив ще създаваме с функцията NewArray, която приема два аргумента – размер на масива и начална стойност на клетките в масива. Параметрите на NewArray са опционални, тоест с NewArray() ще създаваме нов празен масив. Ще считаме, че времевата сложност на NewArray е линейна относно размера на резултатния масив. Също ще считаме, че броя на елементите в масив A (неговия размер) можем да намерим за константно време с израза $|A|$. Индексирането на масив A с размер n ще правим като използваме квадратни скоби, тоест $A[i]$ за $1 \leq i \leq n$ е i-тият елемент на масива A. Ще считаме, че времевата сложност за индексиране на масив е константна. Функцията ArrayAppend ще използваме, когато искаме да добавим елемент към даден масив, така че добавеният елемент да бъде последен, тоест да бъде с най-голям индекс. Тази функция приема два аргумента – масива, към който да бъде добавен елементът, и самият елемент. Времевата сложност на ArrayAppend ще считаме, че е константна амортизирана относно броя на елементите в масива и линейна относно размера на елемента, който бива добавен. Също така ще предполагаме, че паметта, която заема даден масив, е линейна относно сумата на размерите на елементите му.

Навсякъде в псевдокода, където използваме записа за наредена n-орка, ще имаме предвид, че това е масив с n елемента. Тоест всяка наредена n-орка заема памет и се създава за време, колкото е сумата от размерите на елементите ѝ, и предоставя константен достъп до всяка от координатите си.

Нов празен свързан списък ще създаваме за константно време с функцията NewList, която не приема аргументи. Ще считаме, че броя на елементите в свързан списък L можем да намерим за константно време с израза $|L|$. За достъп то първия елемент ще използваме функцията ListHead, която на даден свързан списък, съпоставя указател към първия елемент

5. Разбира се, в допълнение към структурите от данни „естествено число" и „реално число", които имат константна пространствена сложност и върху които ще считаме, че можем да прилагаме основни аритметични операции за константно време.

на списъка за константно време. Ако списъкът е празен, обръщението към ListHead ще считаме, че дава резултат NIL. Изтриване на елемент от свързан списък ще осъществяваме чрез функцията ListRemove, която по даден свързан списък и указател към елемент от него, премахва елемента от списъка за константно време. Функцията ListAppend ще използваме, когато искаме да добавим елемент към края (противоположната страна на ListHead) на даден свързан списък. Тази функция приема два аргумента – свързания списък, към който да бъде добавен елементът, и самият елемент, и дава като резулта указател към вече добавения в списъка елемент. Времевата сложност на ListAppend ще считаме, че е линейна относно размера на елемента, който бива добавен. За да премахнем всички елементи от даден свързан списък ще използваме функцията ListEmpty, която по даден свързан списък изтрива всичките му елементи за време равно на сумата от размерите на елементите в списъка. Ще предполагаме, че можем да изреждаме (в ред започващ от ListHead(L)) елементите на свързан списък L за линейно време относно $|L|$.

Аналогично на NewArray ще използваме функцията NewMatrix, за да създаваме матрици. NewMatrix приема три аргумента – брой редове, брой колони и начална стойност на клетките в матрицата. Ще считаме, че времевата сложност на NewMatrix е линейна относно размера на резултатната матрица. Индексирането на матрица M с размери n на m ще правим като използваме квадратни скоби, тоест $M[i][j]$ за $1 \leq i \leq n$ и $1 \leq j \leq m$ е елементът на M от ред i и колона j. Ще считаме, че времевата сложност за индексиране на матрица е константна.

Нова хеш-таблица ще създаваме с функцията NewHash, която не приема аргументи. Ще считаме, че броя на елементите в хеш-таблица H можем да намерим за константно време с израза $|H|$. За добавяне на елемент в хеш-таблица ще използваме функцията HashInsert, която има три параметъра – първият е хеш-таблицата, в която ще бъде добавен елементът, вторият е ключът на елемента, а третият е стойността на елемента. Ако при изпълняване на обръщението HashInsert(H, k, v), в H вече има елемент с ключ k, ефектът от извикването на функцията ще бъде, че стойността на този елемент ще бъде променена на v. Намирането на стойността на даден елемент, съхранен в хеш-таблица, по неговия ключ ще правим с функцията HashSearch, която приема два аргумента – хеш-таблицата, в която да бъде търсен елементът, и ключът на елемента. Ако няма елемент в хеш-таблицата с търсения ключ, ще считаме, че резултатът от обръщението към HashSearch е NIL. Ще предполагаме, че времевата сложност на функциите HashInsert и HashSearch е линейна относно размера на ключа и размера на стойността на елемента. [6] Също така ще предполагаме, че паметта, която заема дадена хеш-таблица, е линейна относно размера на ключовете и стойностите на елементите, които съхранява. Подобно на свързаните списъци, ще считаме, че можем да итерираме (без значение в какъв ред) елементите на дадена хеш-таблица за линейно време относно техния брой.

Последно, ще спомена, че когато работим с думи или изречения, представени чрез масиви α и β, ще използваме записа $\alpha\beta$, за да означаваме новия масив, представящ конкатенацията на думите или изреченията, представени чрез α и β. Ще считаме, че тази операция се извършва за време $O(|\alpha| + |\beta|)$. Операцията $\alpha[i : j]$ за $1 \leq i \leq j \leq |\alpha|$, създаваща нов масив, състоящ се от елементите $\alpha[i], \alpha[i+1], \ldots, \alpha[j]$, ще считаме, че се извършва за време $O(j - i)$.

6. Тоест, с цел улеснение на анализа на псевдокода, ще игнорираме, че тези операции всъщност имат константна амортизирана сложност и константна сложност в очакване (при определени предположения за входните данни (Cormen и др. 2009)). Представянията на преобразуватели, които ще използваме впоследствие, използват хеш-таблици и поради тази причина не постигат линейната времева сложност при траверсиране, която твърдим, че притежават. Хеш-таблиците в тези представяния могат да бъдат избегнати изцяло, като се използват други структури от данни, и по този начин се постигнат съответните сложности.

2.4 Псевдокод за конструкцията на преобразувател на ниво думи

Сега ще опишем с псевдокод конструкцията на преобразувател на ниво думи за изгладен n-грамен езиков модел.

Алгоритъм 2.1. Намира най-дългия суфикс на дадената дума или изречение, който е с дължина по-малка от n.

```
1: function History(α, n)
2:     return α[max(1, |α| − n + 2) : |α|]
```

С цел удобство в алгоритъм 2.1 дефинираме функцията History, която представя множеството от функции h_n за $n \in \mathbb{N}^+$.

Алгоритъм 2.2. Намира състоянията (и константите e_α за всяко състояние α) на преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1: function WordTransducerStates(P)
2:     Q ← NewHash()
3:     HashInsert(Q, ε, 1)
4:     HashInsert(Q, $, 2)
5:     if |P| > 1 then
6:         for all (d_α, α, e_α) ∈ P[2] do
7:             if HashSearch(Q, α[1]) = NIL then
8:                 HashInsert(Q, α[1], |Q| + 1)
9:     E ← NewArray(|Q|, 0)
10:    for all (d_α, α, e_α) ∈ P[1] do
11:        state = HashSearch(Q, α)
12:        if state ≠ NIL then
13:            E[state] = e_α
14:    for i ← 2, |P| − 1 do
15:        for all (d_α, α, e_α) ∈ P[i] do
16:            if α[i] ≠ $ then
17:                HashInsert(Q, α, |Q| + 1)
18:                ArrayAppend(E, e_α)
19:    return (Q, E)
```

В алгоритъм 2.2 е описана функцията WordTransducerStates, която намира състоянията (и константите e_α за всяко състояние α) на преобразувателя на ниво думи за изгладен n-грамен езиков модел от ARPA описанието на езиковия модел. От определение 2.7 знаем, че състоянията на преобразувателя представляват всички последователности от думи с дължина по-малка от n, които са се срещнали в корпуса, с единственото изключение, че при $n = 1$ трябва да добавим състояние и за $. Можем да забележим, че в конструкцията всички финални състояния са еквивалентни и могат да бъдат заменени с единствено финално състояние (нека го наречем $), което да придобие всички входящи преходи на старите финални състояния. Използвайки това наблюдение, можем да заключим, че за състояния на преобразувателя е достатъчно да вземем $ и всички последователности от думи с дължина по-малка от n, които са се срещнали в корпуса и не завършват на $. От част 1.6 знаем, че

ARPA представянето P на изгладен n-грамен езиков модел съдържа в списъка $P[1]$ по един елемент за всяка дума от речника и думите ^ и \$, а в $P[i]$ за $2 \leq i \leq n$ по един елемент за всяка последователност от думи с дължина i, която се е срещнала в корпуса. Тоест, всички състояния на преобразувателя на ниво думи за езиковия модел, представен чрез P, можем да намерим, като

- вземем ε и \$;
- при $n > 1$ вземем α, ако $(d_\alpha, \alpha, e_\alpha) \in P[1]$ и α е префикс на някое α', такова че $(d_{\alpha'}, \alpha', e_{\alpha'}) \in P[2]$;
- при $n > 1$ вземем α, ако $(d_\alpha, \alpha, e_\alpha) \in P[i]$ за $2 \leq i \leq n-1$ и $\$ \not\geq \alpha$.

Ясно е, че алгоритъм 2.2 прави точно това, като също така подсигурява, че стойностите, които ще бъдат съпоставени на ε и \$ от хеш-таблицата Q, са съответно 1 и 2. Тъй като за всеки елемент на $P[i]$ за $1 \leq i \leq n-1$ се правят константен брой извиквания на функциите HashInsert и HashSearch, времевата сложност на функцията WordTransducerStates е

$$\begin{aligned}
O\left(\sum_{i=1}^{n-1} \sum_{(d_\alpha,\alpha,e_\alpha)\in P[i]} |\alpha|\right) &= O\left(\sum_{i=1}^{n-1} \sum_{(d_\alpha,\alpha,e_\alpha)\in P[i]} i\right) \\
&= O\left(\sum_{i=1}^{n-1} |P[i]| \cdot i\right) \\
&\subseteq O\left(\sum_{i=1}^{n-1} \left(\sum_{j=1}^{m} |\alpha_j|\right) i\right) \\
&\subseteq O\left(\left(\sum_{i=1}^{m} |\alpha_i|\right)\left(\sum_{i=1}^{n-1} i\right)\right) \\
&= O\left(n^2 \sum_{i=1}^{m} |\alpha_i|\right),
\end{aligned}$$

където $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпусът, от който е научен езиковият модел, представен чрез структурата от данни P.

Алгоритъм 2.3. Намира размера на входната азбука на преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1: function AlphabetSize(P)
2:     return |P[1]| − 1
```

От гореспоменатото за ARPA представянето на изгладен n-грамен езиков модел следва, че можем да намерим размера на входната азбука на преобразувателя като извадим единица (заради думата ^) от размера на списъка $P[1]$. Това наблюдение е отразено в алгоритъм 2.3.

В алгоритъм 2.4 е описана функцията WordTransducerTransitions, която намира преходите на преобразувателя на ниво думи за изгладен n-грамен езиков модел, от ARPA описанието на езиковия модел, състоянията на преобразувателя и константите e_α. Конструкцията гласи, че за всяко състояние α и всяка дума a, такива че αa се среща в корпуса или $\alpha = \varepsilon$, $\delta(\alpha, a) = h_n(\alpha a)$ (заради модификацията, която направихме, изключение прави $\delta(\alpha, \$) = \$$) и $\lambda(\alpha, a) = d_{\alpha a}$. Съгласно дефиницията на ARPA формата, всеки елемент от

Алгоритъм 2.4. Намира преходите на преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1: function WordTransducerTransitions(P, Q, E)
2:     alphabet ← AlphabetSize(P)
3:     δ ← NewMatrix(|Q|, alphabet, 0)
4:     λ ← NewMatrix(|Q|, alphabet, 0)
5:     for i ← 1, |P| do
6:         for all (d_α, α, e_α) ∈ P[i] do
7:             if α ≠ ^ then
8:                 source ← HashSearch(Q, α[1 : i − 1])
9:                 target ← 1
10:                if α[i] = $ then
11:                    target ← 2
12:                else if HashSearch(Q, History(α, n)) ≠ NIL then
13:                    target ← HashSearch(Q, History(α, n))
14:                δ[source][α[i]] = target
15:                λ[source][α[i]] = d_α
16:        if i > 1 then
17:            for all (d_α, α, e_α) ∈ P[i − 1] do
18:                source ← HashSearch(Q, α)
19:                if source ≠ NIL and source ≠ $ then
20:                    for a ← 1, alphabet do
21:                        if δ[source][a] = 0 then
22:                            fail ← HashSearch(Q, α[2 : i])
23:                            δ[source][a] = δ[fail][a]
24:                            λ[source][a] = E[fail] · λ[fail][a]
25:    return (δ, λ)
```

списъците в ARPA описанието на изгладен n-грамен езиков модел описва точно един такъв преход, като изключим елемента от $P[1]$, който съответсва на специалната дума ^. Точно това е изразено във **for** цикъла от ред 6 до ред 15. За нефинално състояние α и дума a, такива че αa не се среща в корпуса, конструкцията гласи, че $\delta(\alpha, a) = g_{\mathcal{C}}(h_n(\alpha a)) = g_{\mathcal{C}}(h_n(\alpha' a))$ и $\lambda(\alpha, a) = P^{(|\alpha|+1)}(a \mid \alpha) = e_\alpha \cdot \lambda(\alpha', a)$, където $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$. Тази част от конструкцията е имплементирана в **if** конструкцията от ред 16 до ред 24, в която се разглеждат всички състояния с дължина $i - 1$, които не са получили преход от първия тип. Цикълът от ред 6 добавя по един нов преход на всяка итерация (с изключение на единствената итерация, в която $\alpha =$ ^), а това изисква търсене в хеш-таблицата Q с ключ с дължина по-малка от n. Цикълът от ред 17 прави толкова итерации, колкото е сумата от размерите на списъците $P[i]$ за $1 \leq i < n$, която е не по-голяма от броя на състоянията на преобразувателя. Цикълът от ред 20 прави $O(|Q||\widetilde{\Sigma}|)$ итерации, където $\widetilde{\Sigma}$ е входната азбука на преобразувателя, и всяка от тях изисква търсене в хеш-таблицата Q с ключ с дължина по-малка от n. Следователно времевата сложност на функцията WordTransducerTransitions е $O(n|Q||\widetilde{\Sigma}|)$.

В алгоритъм 2.5 е представена функцията WordTransducer, имплементираща конструкцията от определение 2.7. Резултатът от изпълнението на WordTransducer е преобразувателят на ниво думи за дадения изгладен n-грамен езиков модел. По-конкретно, резултатът се състои от

- s – номерът на началното състояние на преобразувателя;

Алгоритъм 2.5. Построява преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1: function WordTransducer(P)
2:     (Q, E) ← WordTransducerStates(P)
3:     (δ, λ) ← WordTransducerTransitions(P, Q, E)
4:     s ← 1
5:     if |P| > 1 then
6:         s ← HashSearch(Q, ^)
7:     return (s, δ, λ)
```

- δ, λ – матрици с брой редове равен на броя на състоянията на преобразувателя и брой колони равен на големината на входната азбука на преобразувателя, където $\delta[q][a]$ е състоянието, в което преминава преобразувателят при прочитане на дума a от състояние с номер q, а $\lambda[q][a]$ е изходът на този преход.

Тъй като финалното състояние е винаги с номер 2, това представяне на преобразувателя го определя еднозначно с точност до преименуване на състоянията. Така, позовавайки се на твърдение 2.7, получаваме имплементация на конструкцията с времева сложност $O(n^2|\widetilde{\Sigma}|\sum_{i=1}^{m}|\alpha_i|)$ и пространствена сложност $O(n|\widetilde{\Sigma}|\sum_{i=1}^{m}|\alpha_i|)$, където $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпусът, от който е научен езиковият модел.

3 Представяне на езикови модели чрез f-преобразуватели на ниво думи

До този момент показахме как може да бъде построена детерминирана машина, която представя изгладен n-грамен езиков модел. Тази машина е оптимална откъм време за намиране на вероятността, приписана на дадено изречение – тя отговаря на този въпрос за линейно време относно дължината на изречението. Паметта, която заема машината, е от порядъка на сбора на броя състояния и броя преходи, тоест

$$O\left(n\sum_{i=1}^{m}|\alpha_i| + n|\widetilde{\Sigma}|\sum_{i=1}^{m}|\alpha_i|\right) = O\left(n|\widetilde{\Sigma}|\sum_{i=1}^{m}|\alpha_i|\right),$$

където $\widetilde{\Sigma}$ е речникът, а $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпусът над $\widetilde{\Sigma}$ на езиковия модел.

Сега целта ни ще бъде да намалим асимптотично паметта, необходима за представяне на изгладени n-грамни езикови модели, така че тя да бъде от порядъка на $|\widetilde{\Sigma}| + n\sum_{i=1}^{m}|\alpha_i|$. [1] В същото време ще искаме да запазим оптималността на времето за намиране на вероятността, приписана на дадено изречение.

3.1 f-преобразуватели

Тази цел ще постигнем като използваме f-преобразуватели – преобразуватели, позволяващи преходи, които не консумират вход (подобно на ε-преходите при недетерминираните крайни автомати) и които могат да бъдат следвани само когато никой от другите преходи не може да бъде използван. Автомати с f-преходи (така ще наричаме гореописания нов вид преходи) са разглеждани първоначално, с цел редуциране на броя на преходите на краен автомат с фактор големината на азбуката, от Aho и Corasick 1975, въпреки че идеята се среща още в алгоритъма на Knuth-Morris-Pratt (E. Knuth, H. Morris Jr и Pratt 1977), който е измислен през 1970 година.

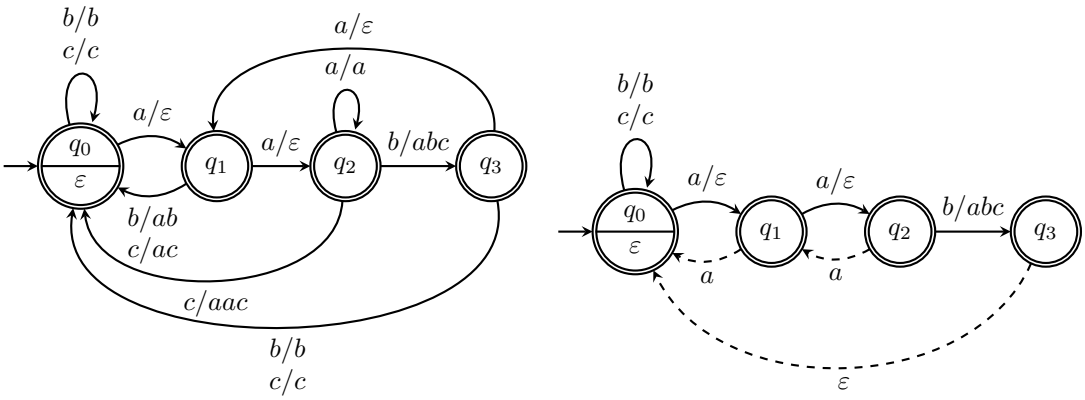
Определение 3.1. *f-преобразувател* е десеторка $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$, където

- Σ е входна азбука;

1. За изгладените n-грамни езикови модели, които се използват на практика, при имплементирането на представянето, което ще разгледаме, може да се използва прост трик, който позволява паметта да не зависи въобще от големината на речника, тоест да бъде $O(n\sum_{i=1}^{m}|\alpha_i|)$.

- $\mathcal{M}$ е моноид;
- Q е крайно множество от състояния;
- $s \in Q$ е начално състояние;
- $F \subseteq Q$ е множество от финални състояния;
- $\delta\colon Q \times \Sigma \to Q$ е частична функция на преходите;
- $f\colon Q \to Q$ е частична функция на f-преходите;
- $\lambda\colon Q \times \Sigma \to M$ е частична функция на изходите;
- $\varphi\colon Q \to M$ е частична функция на f-изходите;
- $\iota \in M$ е начален изход

и $\mathrm{Dom}(\delta) = \mathrm{Dom}(\lambda)$, а $\mathrm{Dom}(f) = \mathrm{Dom}(\varphi)$.



Фигура 3.1: Примери за (а) обикновен преобразувател $\mathcal{T}$ и (б) f-преобразувател $\mathcal{F}$ с изходи от моноида $(\{a, b, c\}^\star, \cdot, \varepsilon)$, където $\cdot$ е операцията „конкатенация на думи".

На фигура 3.1 са представени обикновен преобразувател и f-преобразувател. Новото при f-преобразувателите са f-преходите (представени чрез f-функцията) и техните изходи (представени чрез φ-функцията). Когато изобразяваме f-преобразуватели, ще представяме f-преходите с пунктирани стрелки, над които е изписан изходът на съответния f-преход.

Както вече отбелязахме, f-преобразувателите се траверсират като се следват обикновените преходи (зададени с δ-функцията), а когато няма валиден обикновен преход, се следва f-преходът на текущото състояние, ако такъв има. Тъй като f-преходите не консумират вход, ще дефинираме функциите δ_f и λ_f, които да отразяват поведението на f-преобразувателя при обработката на един входен символ. Тези функции ще задават съответно състоянието, в което преминава f-преобразувателят, и изходът, който той акумулира.

Определение 3.2. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Дефинираме $\delta_f\colon Q \times \Sigma \to Q$ като най-малката относно включване функция $\delta'\colon Q \times \Sigma \to Q$, такава че за $q \in Q$ и $\sigma \in \Sigma$

$$\delta'(q, \sigma) = \begin{cases} \delta(q, \sigma), & \text{ако } !\,\delta(q, \sigma) \\ \delta'(f(q), \sigma), & \text{иначе} \end{cases}$$

Определение 3.3. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Дефинираме $\lambda_f \colon Q \times \Sigma \to M$ като най-малката относно включване функция $\lambda' \colon Q \times \Sigma \to M$, такава че за $q \in Q$ и $\sigma \in \Sigma$

$$\lambda'(q, \sigma) = \begin{cases} \lambda(q, \sigma), & \text{ако } !\,\lambda(q, \sigma) \\ \varphi(q) \cdot \lambda'(f(q), \sigma), & \text{иначе} \end{cases}$$

Поведението на f-преобразувател при обработка на цяла входна дума ще изразим стандартно, разширявайки δ_f и λ_f функциите върху $Q \times \Sigma^\star$.

Определение 3.4. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Дефинираме $\delta_f^\star \colon Q \times \Sigma^\star \to Q$ индуктивно

- $\delta_f^\star(q, \varepsilon) = q$ за $q \in Q$;
- $\delta_f^\star(q, \alpha a) = \delta_f(\delta_f^\star(q, \alpha), a)$ за всяко $q \in Q$, $\alpha \in \Sigma^\star$ и $a \in \Sigma$.

Определение 3.5. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Дефинираме $\lambda_f^\star \colon Q \times \Sigma^\star \to M$ индуктивно

- $\lambda_f^\star(q, \varepsilon) = e$ за $q \in Q$;
- $\lambda_f^\star(q, \alpha a) = \lambda_f^\star(q, \alpha) \cdot \lambda_f(\delta_f^\star(q, \alpha), a)$ за всяко $q \in Q$, $\alpha \in \Sigma^\star$ и $a \in \Sigma$.

Както при обикновените преобразователи, ще въведем функция $O_{\mathcal{F}}$, отразяваща резултата от траверсиране на f-преобразувателя $\mathcal{F}$ с дадена входна дума.

Определение 3.6. Нека $\mathcal{F} = (\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. $O_{\mathcal{T}} \colon \Sigma^\star \to M$, дефинирана за $\alpha \in \Sigma^\star$ така

$$O_{\mathcal{T}}(\alpha) = \begin{cases} \iota \cdot \lambda_f^\star(s, \alpha), & \text{ако } \delta_f^\star(s, \alpha) \in F \\ \neg\,!, & \text{иначе} \end{cases}$$

наричаме *функцията, представена от f-преобразувателя $\mathcal{F}$*.

На примерите от фигура 3.1 преобразувателят $\mathcal{T}$ и f-преобразувателят $\mathcal{F}$ представят функцията от $\{a, b, c\}^\star$ в $\{a, b, c\}^\star$, съпоставяща на всяка дума $\alpha \in \{a, b, c\}^\star$ думата, получена от α чрез заместване на всяко срещане на aab с abc. Можем да забележим как f-преходите ни позволяват да редуцираме броя на δ-преходите. Например, тъй като на фигура 3.1(а) състоянието q_3 има същите преходи като състоянието q_0, на фигура 3.1(б) неговите преходи са заменени с един единствен f-преход. Също така преходите с b и c на състоянието q_1 от фигура 3.1(а) са заменени на фигура 3.1(б) с един единствен f-преход с изход a към състоянието q_0, тъй като q_0 има същите преходи с b и c като q_1 с единствената разлика, че изходите на преходите на q_1 имат конкатениран отляво допълнителен символ a.

Аналогично на ситуацията с преобразувателите, началните изходи не увеличават съществено изразителната способност на f-преобразувателите. Тоест за всеки f-преобразувател съществува f-преобразувател с единичен начален изход, който е еквивалентен на изходния с точност до ε.

Твърдение 3.1. *Нека $\mathcal{F} = (\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Тогава f-преобразувателят $\mathcal{F}' = (\Sigma, \mathcal{M}, Q', s', F, \delta', f', \lambda', \varphi', e)$, където*

- $Q' = Q \cup \{s'\}$;
- $s' \notin Q$;

- *за* $q \in Q'$ *и* $\sigma \in \Sigma$

$$\delta'(q,\sigma) = \begin{cases} \delta(q,\sigma), & \text{ако } q \in Q \\ \delta(s,\sigma), & \text{ако } q = s' \end{cases} \qquad \lambda'(q,\sigma) = \begin{cases} \lambda(q,\sigma), & \text{ако } q \in Q \\ \iota \cdot \lambda(s,\sigma), & \text{ако } q = s' \end{cases}$$

- *за* $q \in Q'$

$$f'(q) = \begin{cases} f(q), & \text{ако } q \in Q \\ f(s), & \text{ако } q = s' \end{cases} \qquad \varphi'(q) = \begin{cases} \varphi(q), & \text{ако } q \in Q \\ \iota \cdot \varphi(q), & \text{ако } q = s' \end{cases}$$

е такъв, че

$$(\forall \alpha \in \Sigma^+)(O_{\mathcal{F}}(\alpha) = O_{\mathcal{F}'}(\alpha)).$$

Доказателство. Тъй като за всяко $q \in Q$ и $\sigma \in \Sigma$ е вярно, че $\delta'(q,\sigma) = \delta(q,\sigma)$ и $f'(q) = f(q)$, то е в сила, че

$$(\forall q \in Q)(\forall \sigma \in \Sigma)(\delta'_{f'}(q,\sigma) = \delta_f(q,\sigma) \wedge \lambda'_{f'}(q,\sigma) = \lambda_f(q,\sigma)).$$

С индукция по $|\alpha|$ ще докажем, че

$$(\forall \alpha \in \Sigma^+)(\delta'^{\star}_{f'}(s',\alpha) = \delta^{\star}_f(s,\alpha) \wedge \lambda'^{\star}_{f'}(s',\alpha) = \iota \cdot \lambda^{\star}_f(s,\alpha)).$$

$|\alpha| = 1$

Тогава $\alpha \in \Sigma$ и има два случая. Нека първо да разгледаме случая, в който $!\,\delta(s,\alpha)$, тоест $!\,\delta'(s',\alpha)$.

1. $\delta'^{\star}_{f'}(s',\alpha) = \delta'_{f'}(s',\alpha) = \delta'(s',\alpha) = \delta(s,\alpha) = \delta_f(s,\alpha) = \delta^{\star}_f(s,\alpha)$;
2. $\lambda'^{\star}_{f'}(s',\alpha) = \lambda'_{f'}(s',\alpha) = \lambda'(s',\alpha) = \iota \cdot \lambda(s,\alpha) = \iota \cdot \lambda_f(s,\alpha) = \iota \cdot \lambda^{\star}_f(s,\alpha)$.

Нека сега да разгледаме случая, в който $\neg\,!\,\delta(s,\alpha)$, тоест $\neg\,!\,\delta'(s',\alpha)$.

1. $\begin{aligned}\delta'^{\star}_{f'}(s',\alpha) &= \delta'_{f'}(s',\alpha) = \delta'_{f'}(f'(s'),\alpha) = \delta'_{f'}(f(s),\alpha) = \delta_f(f(s),\alpha) \\ &= \delta_f(s,\alpha) = \delta^{\star}_f(s,\alpha);\end{aligned}$
2. $\begin{aligned}\lambda'^{\star}_{f'}(s',\alpha) &= \lambda'_{f'}(s',\alpha) = \varphi'(s') \cdot \lambda'_{f'}(f'(s'),\alpha) = \varphi(s) \cdot \lambda'_{f'}(f(s),\alpha) \\ &= \varphi(s) \cdot \lambda_f(f(s),\alpha) = \lambda_f(s,\alpha) = \lambda^{\star}_f(s,\alpha).\end{aligned}$

$|\alpha| \rightsquigarrow |\alpha| + 1$

Нека $\alpha = \alpha' a$ и твърдението е в сила за всяка дума с дължина $|\alpha'|$. Тогава

1. $\begin{aligned}\delta'^{\star}_{f'}(s',\alpha) &= \delta'^{\star}_{f'}(s',\alpha' a) = \delta'_{f'}(\delta'^{\star}_{f'}(s',\alpha'),a) = \delta'_{f'}(\delta^{\star}_f(s,\alpha'),a) \\ &= \delta_f(\delta^{\star}_f(s,\alpha'),a) = \delta^{\star}_f(s,\alpha' a) = \delta^{\star}_f(s,\alpha);\end{aligned}$
2. $\begin{aligned}\lambda'^{\star}_{f'}(s',\alpha) &= \lambda'^{\star}_{f'}(s',\alpha' a) = \lambda'^{\star}_{f'}(s',\alpha') \cdot \lambda'_{f'}(\delta'^{\star}_{f'}(s',\alpha'),a) \\ &= \iota \cdot \lambda^{\star}_f(s,\alpha') \cdot \lambda'_{f'}(\delta^{\star}_f(s,\alpha'),a) = \iota \cdot \lambda^{\star}_f(s,\alpha') \cdot \lambda_f(\delta^{\star}_f(s,\alpha'),a) \\ &= \iota \cdot \lambda^{\star}_f(s,\alpha' a) = \iota \cdot \lambda^{\star}_f(s,\alpha).\end{aligned}$

Сега за всяка дума $\alpha \in \Sigma^+$ е в сила, че

$$\alpha \in \mathrm{Dom}(O_{\mathcal{F}}) \iff \delta^{\star}_f(s,\alpha) \in F \iff \delta'^{\star}_{f'}(s',\alpha) \in F \iff \alpha \in \mathrm{Dom}(O_{\mathcal{F}'})$$

и ако $\alpha \in \mathrm{Dom}(O_{\mathcal{F}})$, то

$$O_{\mathcal{F}}(\alpha) = \iota \cdot \lambda^{\star}_f(s,\alpha) = \lambda'^{\star}_{f'}(s',\alpha) = e \cdot \lambda'^{\star}_{f'}(s',\alpha) = O_{\mathcal{F}'}(\alpha). \qquad \square$$

Причината да позволяваме f-преобразувателите да имат начални изходи е улеснение при въвеждане на понятието „каноничен f-преобразувател".

Преобразувателите без f-преходи от предходната глава са частен случай на f-преобразувателите от текущата. Въпреки това позволяването на f-преходи не увеличава тяхната изразителност.

Определение 3.7. *Подлежащият преобразувател* на f-преобразувателя $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ наричаме преобразувателя $(\Sigma, \mathcal{M}, Q, s, F, \delta_f, \lambda_f, \iota)$.

От дефинициите на функция, представена от преобразувател, и функция, представена от f-преобразувател, става ясно, че даден f-преобразувател и неговият подлежащ преобразувател представят една и съща функция.

Твърдение 3.2. *Нека $\mathcal{F}$ е f-преобразувател и $\mathcal{T}$ е подлежащият му преобразувател. Тогава $O_{\mathcal{F}} = O_{\mathcal{T}}$.* □

За разлика от преобразувателите, f-преобразувателите не гарантират обработката на дадена входна дума за линейно време (брой направени δ-преходи и f-преходи) относно дължината ѝ. Затова за всеки построен в тази глава f-преобразувател ще се налага да доказваме, че притежава това свойство. С цел формализиране на тези доказателства ще дефинираме две фукнции $\#_\delta$ и $\#_f$, които да отразяват съответно броя δ-преходи и f-преходи, които даден f-преобразувател прави при обработката на дадена входна дума.

Определение 3.8. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Дефинираме $\#_\delta \colon Q \times \Sigma^\star \to \mathbb{N}$ като най-малката относно включване функция $\#'_\delta \colon Q \times \Sigma^\star \to \mathbb{N}$, такава че за $q \in Q$ и $a \in \Sigma$

$$\#'_\delta(q, a) = \begin{cases} 1, & \text{ако } !\,\delta(q, a) \\ \#'_\delta(f(q), a), & \text{иначе} \end{cases}$$

и за $q \in Q$ и $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \dots a_n$

$$\#'_\delta(q, \alpha) = \sum_{i=1}^{n} \#'_\delta(\delta_f^\star(q, a_1 a_2 \dots a_{i-1}), a_i).$$

От определение 3.8 става ясно, че за всяка дума α, такава че $!\,\delta_f^\star(q, \alpha)$, е в сила, че $\#_\delta(q, \alpha) = |\alpha|$.

Определение 3.9. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Функцията $\#_f \colon Q \times \Sigma^\star \to \mathbb{N}$ дефинираме за $q \in Q$ и $a \in \Sigma$ така

$$\#_f(q, a) = \begin{cases} 0, & \text{ако } !\,\delta(q, a) \\ \#_f(f(q), a) + 1, & \text{иначе} \end{cases}$$

и за $q \in Q$ и $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \dots a_n$ така

$$\#_f(q, \alpha) = \sum_{i=1}^{n} \#_f(\delta_f^\star(q, a_1 a_2 \dots a_{i-1}), a_i).$$

В тази дефиниция не е нужно явно да изискваме $\#_f$ да бъде най-малката относно включване функция, удовлетворяваща тези условия, тъй като в случая съществува единствена такава функция.

f-преобразувателите не консумират входна буква, когато следват f-преход. Поради тази причина доказателствата на твърденията, описващи поведението на f-преобразувател при обработка на дадена входна дума, ще правим като провеждаме индукция не само по дължината на входната дума, но и по някаква друга характеристика на състоянията, обвързана с f-преходите им. Следващите понятия въвеждаме точно с тази цел.

Определение 3.10. Нека $\mathcal{F} = (\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. *f-граф* на $\mathcal{F}$ наричаме графа $G_f = (Q, E_f)$, където

$$E_f = \{(q_1, q_2) \in Q \times Q \mid f(q_1) = q_2\}.$$

Определение 3.11. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Функцията $\mathrm{level}_f \colon Q \to \mathbb{N}$ дефинираме за $q \in Q$ така

$$\mathrm{level}_f(q) = \begin{cases} 0, & \text{ако } \neg\,!\, f(q) \\ \mathrm{level}_f(f(q)) + 1, & \text{иначе} \end{cases}$$

В тази дефиниция също не се налага да изискваме минималност относно включване.

Твърдение 3.3. *Нека $\mathcal{F} = (\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател, чийто f-граф G_f е ацикличен. Тогава функцията* level_f *е дефинирана за всяко състояние на $\mathcal{F}$.*

Доказателство. Да допуснем, че level_f не е дефинирана за $q \in Q$. Да разгледаме редицата от състояния на $\mathcal{F}$

$$f^{(0)}(q), f^{(1)}(q), f^{(2)}(q), \ldots, f^{(n)}(q), \ldots,$$

където $f^{(0)}$ е идентитетът от Q в Q, а $f^{(n)}$ за $n \in \mathbb{N}^+$ е композицията на f със себе си n пъти. Ясно е, че редицата е крайна, тъй като G_f е ацикличен, и последният ѝ елемент $f^{(k)}(q)$ е такъв, че $\neg\,!\, f^{(k+1)}(q)$. Следователно с индукция по i лесно се вижда, че

$$(\forall 0 \leq i \leq k)(\mathrm{level}_f(f^{(k-i)}(q)) = i).$$

Но това противоречи с допуснатото. Следователно $\mathrm{Dom}(\mathrm{level}_f) = Q$. □

3.2 f-преобразувател на ниво думи за езиков модел

Както вече казахме, искаме, използвайки f-преобразуватели, да получим представяне на изгладени n-грамни езикови модели, изискващо памет от порядъка на $|\widetilde{\Sigma}| + n\sum_{i=1}^{m} |\alpha_i|$ и постигащо линейна времева сложност за намиране на вероятността на дадено изречение. Това ще постигнем, като се възползваме от наблюдение, което направихме в твърдение 1.6, а именно, че не се нуждаем от стойностите на параметрите, за които няма свидетели в корпуса, тоест не е нужно да ги кодираме експлицитно. Конструкцията от определение 2.7 се възползва частично от тази информация, като добавя състояния само за тези истории, които са се срещнали в корпуса, и така елиминира голяма част от преходите, които биха съответствали на параметри без свидетел в корпуса.

Нека $\widetilde{\Sigma}$ е речник и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. Нека да разгледаме преобразувателя на ниво думи $\mathcal{T}$ за изгладен $(n+1)$-грамен езиков модел $P^{(n+1)}$ за корпуса $\mathcal{C}$. Неговите състояния съответстват на всички последователности от думи с дължина ненадминаваща n, които са се срещнали в корпуса. Също така всяко едно негово нефинално състояние α има преход с вход a и изход $P^{(|\alpha|+1)}(a \mid \alpha)$ за всяка дума $a \in \widetilde{\Sigma} \cup \{\$\}$. Да разгледаме едно такова състояние

α. От дефиницията на изгладен $(n+1)$-грамен езиков модел за $\mathcal{C}$ следва, че за всяка дума $a \in \widetilde{\Sigma} \cup \{\$\}$, такава че $\#_{\mathcal{C}}(\alpha a) = 0$,

$$P^{(|\alpha|+1)}(a \mid \alpha) = e_\alpha P^{(|\alpha'|+1)}(a \mid \alpha'),$$

където $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$. Тоест за всички такива думи a има една и съща константа e_α, а множителят $P^{(|\alpha'|+1)}(a \mid \alpha')$ може да бъде намерен, траверсирайки преобразувателя от състояние α' с дума a. Това наблюдение мотивира използването на f-преобразуватели, тъй като те позволяват да кодираме с един f-преход (от α до α' с изход e_α) всички преходи в преобразувателя $\mathcal{T}$ от α с дума a, такава че $\#_{\mathcal{C}}(\alpha a) = 0$. Тъй като параметрите на изгладените 1-грамни езикови модели нямат константи e_α, в конструкцията, която ще покажем, състоянието ε няма да има f-преход, а ще има същите δ-преходи като в преобразувателя $\mathcal{T}$.

Следвайки гореописаната интуиция, f-преобразувателите, с които ще представяме n-грамни езикови модели, ще имат изходи от моноида $(\mathbb{R}_{\geq 0}, \cdot, 1)$, който ще означаваме с $\mathcal{R}_{\geq 0}$, а не от $\mathcal{R}_{[0,1]}$. Това се налага поради свободата, която дефиницията на изгладен n-грамен езиков модел дава на константите e_α – те могат да бъдат произволни неотрицателни реални числа.

Определение 3.12. Нека $\widetilde{\Sigma}$ е азбука и $\mathcal{C}$ е корпус над $\widetilde{\Sigma}$. Нека $P^{(n)}$ е изгладен n-грамен езиков модел за корпуса $\mathcal{C}$. Тогава f-преобразувателя

$n = 1$
$(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \varnothing, \lambda, \varnothing, \iota)$, където $(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}, Q, s, F, \delta, \lambda, \iota)$ е преобразувателят на ниво думи за $P^{(n)}$;

$n \geq 2$
$(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, \hat{}, F, \delta, f, \lambda, \varphi, 1)$, където $(\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, \hat{}, F, \delta', \lambda', 1)$ е преобразувателят на ниво думи за $P^{(n)}$ и

- за $\alpha \in Q$ и $a \in \widetilde{\Sigma} \cup \{\$\}$

$$\delta(\alpha, a) = \begin{cases} \delta'(\alpha, a), & \text{ако } \#_{\mathcal{C}}(\alpha a) > 0 \vee \alpha = \varepsilon \\ \neg\,!, & \text{иначе} \end{cases} \qquad \lambda(\alpha, a) = \begin{cases} \lambda'(\alpha, a), & \text{ако } !\,\delta(\alpha, a) \\ \neg\,!, & \text{иначе} \end{cases}$$

- за $\alpha \in Q$

$$f(\alpha) = \begin{cases} \alpha', & \text{ако } \alpha \notin F \wedge (\exists a \in \widetilde{\Sigma} \cup \{\$\})(\neg\,!\,\delta(\alpha, a)) \\ \neg\,!, & \text{иначе} \end{cases} \qquad \varphi(\alpha) = \begin{cases} e_\alpha, & \text{ако } !\,f(\alpha) \\ \neg\,!, & \text{иначе} \end{cases}$$

където $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$,

наричаме *f-преобразувателят на ниво думи за $P^{(n)}$*.

Конструкцията, която описахме, е представена от Allauzen, Mohri и Roark 2003. Тук ще докажем нейната коректност, тоест че построеният f-преобразувател представя съответния изгладен n-грамен езиков модел, тъй като това не е направено в статията.

Твърдение 3.4. *Нека $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Тогава f-графът $G_f = (Q, E_f)$ на $\mathcal{F}$ е ацикличен.*

Доказателство. Съгласно дефиницията на f, за ребро $(\alpha, \beta) \in E_f$ в G_f е в сила, че $|\alpha| > |\beta|$ и по-точно, че $|\alpha| = |\beta| + 1$. Ако допуснем, че в G_f има цикъл $\alpha_1, \alpha_2, \ldots, \alpha_m, \alpha_1$, то би следвало, че $|\alpha_1| > |\alpha_1|$, което е абсурд. Следователно G_f е ацикличен. □

Това означава, че функцията level_f на f-преобразувателя на ниво думи $\mathcal{F}$ за изгладен n-грамен езиков модел е дефинирана за всяко негово състояние.

За да докажем коректността на конструкцията, ще покажем, че f-преобразувателят на ниво думи за изгладен n-грамен езиков модел има същото поведение като преобразувателя на ниво думи за същия езиков модел.

Твърдение 3.5. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$. Нека $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$, $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за $P^{(n)}$, а $\mathcal{T} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q', s', F', \delta', \lambda', \iota')$ е преобразувателят на ниво думи за $P^{(n)}$. Тогава за всяко $\alpha \in Q \setminus F$ и всяко $a \in \widetilde{\Sigma} \cup \{\$\}$ е в сила, че*

$$\delta_f(\alpha, a) = \delta'(\alpha, a) \wedge \lambda_f(\alpha, a) = \lambda'(\alpha, a).$$

Доказателство. За $n = 1$ твърдението следва директно от дефиницията на $\mathcal{F}$, тъй като тогава $\delta_f = \delta = \delta'$ и $\lambda_f = \lambda = \lambda'$.

За $n \geq 2$ от определение 2.7 следва, че $Q = Q'$ и $F = F'$. В този случай твърдението ще докажем с индукция по $\text{level}_f(\alpha)$.

$\text{level}_f(\alpha) = 0$

Тогава $\neg\,!\, f(\alpha)$, тоест $(\forall a \in \widetilde{\Sigma} \cup \{\$\})(!\, \delta(\alpha, a))$. Тогава твърдението следва от дефинициите на δ и λ, тъй като за $a \in \widetilde{\Sigma} \cup \{\$\}$

$$\delta_f(\alpha, a) = \delta(\alpha, a) = \delta'(\alpha, a),$$
$$\lambda_f(\alpha, a) = \lambda(\alpha, a) = \lambda'(\alpha, a).$$

$\text{level}_f(\alpha) \rightsquigarrow \text{level}_f(\alpha) + 1$

Нека $\alpha \in Q \setminus F$, $\text{level}_f(\alpha) > 0$ и твърдението е в сила за всяко състояние $\beta \in Q \setminus F$, такова че $\text{level}_f(\beta) = \text{level}_f(\alpha) - 1$. Нека $a \in \widetilde{\Sigma} \cup \{\$\}$.

1сл. $\#_{\mathcal{C}}(\alpha a) > 0$

Тогава можем да повторим разсъжденията от базовия случай.

2сл. $\#_{\mathcal{C}}(\alpha a) = 0 \wedge \alpha \neq \varepsilon$

Следователно $\neg\,!\, \delta(\alpha, a)$, $!\, f(\alpha)$ и

1. $\delta_f(\alpha, a) = \delta_f(f(\alpha), a) = \delta'(f(\alpha), a) = g_{\mathcal{C}}(h_n(f(\alpha)a)) = g_{\mathcal{C}}(h_n(\alpha a)) = \delta'(\alpha, a)$;
   Предпоследното равенство е вярно, тъй като ако $|\alpha a| = n$, то $|f(\alpha)a| = n - 1$ и $h_n(\alpha a) = f(\alpha)a = h_n(f(\alpha)a)$. Ако $|\alpha a| < n$, то $|f(\alpha)a| < n$, $h_n(\alpha a) = \alpha a$ и $h_n(f(\alpha)a) = f(\alpha)a$. Но щом $\#_{\mathcal{C}}(\alpha a) = 0$, то
   $$g_{\mathcal{C}}(h_n(\alpha a)) = g_{\mathcal{C}}(\alpha a) = g_{\mathcal{C}}(f(\alpha)a) = g_{\mathcal{C}}(h_n(f(\alpha)a)).$$
2. $\lambda_f(\alpha, a) = \varphi(\alpha) \cdot \lambda_f(f(\alpha), a) = \varphi(\alpha) \cdot \lambda'(f(\alpha), a) = \varphi(\alpha) \cdot P^{(|f(\alpha)|+1)}(a \mid f(\alpha))$
   $= P^{(|\alpha|+1)}(a \mid \alpha) = \lambda'(\alpha, a)$.

Случаят $\#_{\mathcal{C}}(\alpha a) = 0 \wedge \alpha = \varepsilon$ е невъзможен, тъй като $\text{level}_f(\varepsilon) = 0$. □

Сега, знаейки че f-преобразувателите от определение 3.12 се държат като преобразувателите от определение 2.7 за същите изгладени n-грамни езикови модели, можем да заключим, че f-преобразувателите от определение 3.12 наистина представят съответните им изгладени n-грамни езикови модели.

Твърдение 3.6. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава за f-преобразувателя на ниво думи $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ за $P^{(n)}$ е в сила, че*

- $\mathrm{Dom}(O_{\mathcal{F}}) = \widetilde{\Sigma}^{\star} \circ \{\$\}$;
- $(\forall \alpha \in \widetilde{\Sigma}^{\star})(O_{\mathcal{F}}(\alpha\$) = P^{(n)}(\{\hat{}\,\alpha\$\}))$.

Доказателство. Нека $\mathcal{T} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q', s', F', \delta', \lambda', \iota')$ е преобразувателят на ниво думи за $P^{(n)}$. Ясно е, че $Q = Q'$, $s = s' = h_n(\hat{})$, $F = F'$ и $\iota = \iota' = 1$. Тъй като функциите δ_f, λ_f, δ' и λ' не са дефинирани за никое финално състояние, от твърдение 3.5 следва, че $\delta_f^{\star} = \delta'^{\star}$ и $\lambda_f^{\star} = \lambda'^{\star}$. Тогава

1. за всяко $\alpha \in (\widetilde{\Sigma} \cup \{\$\})^{\star}$

$$\alpha \in \mathrm{Dom}(O_{\mathcal{F}}) \iff \delta_f^{\star}(s, \alpha) \in F \iff \delta'^{\star}(s', \alpha) \in F' \iff \alpha \in \mathrm{Dom}(O_{\mathcal{T}}) = \widetilde{\Sigma}^{\star} \circ \{\$\};$$

2. за всяко $\alpha \in \widetilde{\Sigma}^{\star}$

$$O_{\mathcal{F}}(\alpha\$) = \iota \cdot \lambda_f^{\star}(s, \alpha\$) = \iota' \cdot \lambda'^{\star}(s', \alpha\$) = O_{\mathcal{T}}(\alpha\$) = P^{(n)}(\{\hat{}\,\alpha\$\}). \qquad \square$$

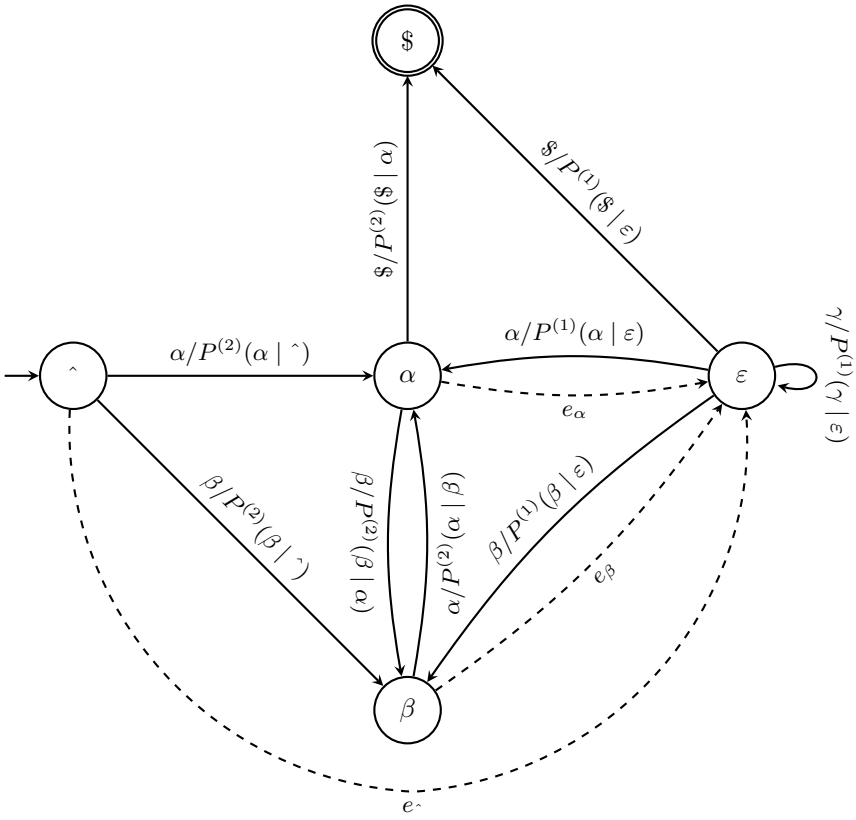
На фигура 3.2 е изобразен f-преобразувателят на ниво думи за $P^{(2)}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\,\alpha\beta\alpha\$$, $\hat{}\,\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$. Можем, сравнявайки с фигура 2.2, да забележим как f-преходите ни позволяват да намалим броя на нужните за представянето на езиковия модел от примера преходи.

Дефинираните в определение 3.12 f-преобразуватели имат размер, зависещ мултилинейно от n и броя думи в корпуса и линейно от броя думи в речника. [2] Тъй като $|\,\mathrm{Dom}(f)| \leq |Q|$, когато оценяваме размера на един f-преобразувател, ще се интересуваме само от броя на неговите състояния и δ-преходи.

Твърдение 3.7. *Нека $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава f-преобразувателят на ниво думи $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ за $P^{(n)}$ е такъв, че*

- $|Q| \in O(n \sum_{i=1}^{m} |\alpha_i|)$;
- $|\,\mathrm{Dom}(\delta)| \in O(|\widetilde{\Sigma}| + n \sum_{i=1}^{m} |\alpha_i|)$.

2. В конструкцията от определение 3.12 можем да забележим, че изискваме единствено началното състояние s да има преход с всяка дума от речника. Имплементирайки конструкцията, можем да кодираме явно само изходящите от s преходи с думи a, за които $\#_{\mathcal{C}}(sa) > 0$. За останалите преходи, излизащи от s, знаем, че те влизат в s. Също така в повечето изгладени n-грамни езикови модели, използвани на практика, тези преходи имат един и същ изход и затова можем да ги имплементираме с проста условна конструкция. Така от разсъжденията в доказателството на твърдение 3.7, следва, че в тези случаи можем да имплементираме конструкцията от определение 3.12 с пространствена сложност, която не зависи от броя думи в речника.



Фигура 3.2: f-преобразувателят на ниво думи за $P^{(2)}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\alpha\beta\alpha\$, \hat{}\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$.

Доказателство. Нека $\mathcal{T}$ е преобразувателят на ниво думи за езиковия модел $P^{(n)}$. Състоянията на $\mathcal{F}$ са същите като състоянията на $\mathcal{T}$. От твърдение 2.7 знаем, че те са $O(n\sum_{i=1}^{m}|\alpha_i|)$.

При $n = 1$ $\mathcal{F}$ има същите δ-преходи като $\mathcal{T}$. Тогава от разсъжденията в доказателството на твърдение 2.7 следва, че

$$|\operatorname{Dom}(\delta)| = |\widetilde{\Sigma}| + 1 \in O(|\widetilde{\Sigma}| + n\sum_{i=1}^{m}|\alpha_i|).$$

При $n \geq 2$ всяко нефинално състояние α има изходящ преход с буква a от $\widetilde{\Sigma} \cup \{\$\}$ само ако $\alpha = \varepsilon$ или $\#_{\mathcal{C}}(\alpha a) > 0$. Тоест

$$\begin{aligned}
|\operatorname{Dom}(\delta)| &= |\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid !\,\delta(\alpha, a)\}| \\
&= |\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid \alpha = \varepsilon \vee \#_{\mathcal{C}}(\alpha a) > 0\}| \\
&\leq |\{(\varepsilon, a) \mid a \in \widetilde{\Sigma} \cup \{\$\}\}| + |\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid \#_{\mathcal{C}}(\alpha a) > 0\}| \\
&= |\widetilde{\Sigma}| + 1 + |\bigcup_{j=0}^{n-1}\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid |\alpha| = j \wedge \#_{\mathcal{C}}(\alpha a) > 0\}|
\end{aligned}$$

$$
\begin{aligned}
&= |\widetilde{\Sigma}| + 1 + |\bigcup_{j=0}^{n-1} \{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid |\alpha| = j \wedge \sum_{i=1}^{m} \#_{\alpha_i}(\alpha a) > 0\}| \\
&= |\widetilde{\Sigma}| + 1 + |\bigcup_{i=1}^{m} \bigcup_{j=0}^{n-1} \{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid |\alpha| = j \wedge \#_{\alpha_i}(\alpha a) > 0\}| \\
&\leq |\widetilde{\Sigma}| + 1 + \sum_{i=1}^{m} \sum_{j=0}^{n-1} |\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid |\alpha| = j \wedge \#_{\alpha_i}(\alpha a) > 0\}| \\
&\leq |\widetilde{\Sigma}| + 1 + \sum_{i=1}^{m} \sum_{j=0}^{n-1} |\alpha_i| \\
&= |\widetilde{\Sigma}| + 1 + n \sum_{i=1}^{m} |\alpha_i| \in O(|\widetilde{\Sigma}| + n \sum_{i=1}^{m} |\alpha_i|).
\end{aligned}
$$

В последното неравенство използваме, че за $j \in \mathbb{N}$ и $1 \leq i \leq m$

$$\{(\alpha, a) \in Q \times (\widetilde{\Sigma} \cup \{\$\}) \mid |\alpha| = j \wedge \#_{\alpha_i}(\alpha a) > 0\} \subseteq \{\alpha \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^{j+1} \mid \#_{\alpha_i}(\alpha)\},$$

за което знаем от доказателството на твърдение 2.7, че има не повече от $|\alpha_j|$ елемента. □

За разлика от преобразувателите, които правят точно толкова прехода, колкото е дължината на входа, f-преобразувателите могат да правят сумарно повече δ-преходи и f-преходи. Сега ще покажем, че f-преобразувателите на ниво думи за изгладени n-грамни езикови модели, могат да бъдат траверсирани за линейно време относно дължината на входното изречение. По-точно, ще покажем, че δ-преходите и f-преходите, които те правят, са не повече от два пъти броя на думите във входното изречение.

Твърдение 3.8. *Нека $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Тогава за всяко $\alpha \in \widetilde{\Sigma}^{\star} \circ \{\$\}$ е в сила, че*

$$\#_{\delta}(s, \alpha) + \#_{f}(s, \alpha) \in O(|\alpha|).$$

Доказателство. Първо, с индукция по $\mathrm{level}_f(q)$ ще докажем, че

$$(\forall q \in Q \setminus F)(\forall a \in \widetilde{\Sigma} \cup \{\$\})(\#_f(q, a) + |\delta_f(q, a)| - |q| \leq \#_{\delta}(q, a)).$$

$\mathbf{level_f(q) = 0}$

Тогава $!\,\delta(q, a)$ за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$ и следователно $\#_f(q, a) = 0$, $\#_{\delta}(q, a) = 1$ и $|\delta_f(q, a)| = |\delta(q, a)| \leq |q| + 1$. Тоест твърдението е в сила.

$\mathbf{level_f(q) \rightsquigarrow level_f(q) + 1}$

Нека $q \in Q \setminus F$, $\mathrm{level}_f(q) > 0$ и твърдението е в сила за всяко състояние $p \in Q \setminus F$, такова че $\mathrm{level}_f(p) = \mathrm{level}_f(q) - 1$. Нека $a \in \widetilde{\Sigma} \cup \{\$\}$.

1. Ако $!\,\delta(q, a)$, разсъждението е същото като в базовия случай.
2. Ако $\neg !\,\delta(q, a)$, то $|q| > 0$, $|q| = |f(q)| + 1$ и

$$
\begin{aligned}
\#_f(q, a) + |\delta_f(q, a)| - |q| &= \#_f(f(q), a) + 1 + |\delta_f(f(q), a)| - (|f(q)| + 1) \\
&= \#_f(f(q), a) + |\delta_f(f(q), a)| - |f(q)| \\
&\leq \#_{\delta}(f(q), a) \\
&= \#_{\delta}(q, a).
\end{aligned}
$$

Нека сега $\alpha \in \widetilde{\Sigma}^\star \circ \{\$\}$ и $\alpha = a_1 a_2 \dots a_{m+1}$. Нека $q_0, q_1, q_2, \dots, q_{m+1}$ е редицата от състояния, за която $q_0 = s$ и $q_i = \delta_f^\star(s, a_1 a_2 \dots a_i)$ за $1 \leq i \leq m+1$. Ясно е, че само $q_{m+1} \in F$, което означава, че съгласно твърдението, което доказахме,

$$(\forall 1 \leq i \leq m+1)(\#_f(q_{i-1}, a_i) + |q_i| - |q_{i-1}| \leq \#_\delta(q_{i-1}, a_i)).$$

Сега, сумирайки по i, получаваме

$$\sum_{i=1}^{m+1} \#_f(q_{i-1}, a_i) + \sum_{i=1}^{m+1} (|q_i| - |q_{i-1}|) \leq \sum_{i=1}^{m+1} \#_\delta(q_{i-1}, a_i)$$
$$\#_f(s, \alpha) + |q_{m+1}| - |s| \leq \#_\delta(s, \alpha).$$

Тоест броят на направените f-преходи е не по-голям от броя на направените δ-преходи

$$\#_f(s, \alpha) \leq \#_\delta(s, \alpha) + |s| - |q_{m+1}| \leq \#_\delta(s, \alpha) + 1 - |q_{m+1}| \leq \#_\delta(s, \alpha),$$

от което следва, че

$$\#_f(s, \alpha) + \#_\delta(s, \alpha) \leq 2\#_\delta(s, \alpha) \in O(|\alpha|). \qquad \square$$

Важно свойство на f-преобразувателите на ниво думи за изгладени n-грамни езикови модели е, че ако дадено състояние α на f-преобразувателя има δ-преход с дума a от речника, то всяко състояние суфикс на α също ще има δ-преход с тази дума.

Със Suf означаваме функцията, която съпоставя на съвкупност от изречения множеството от всички техни суфикси.

Определение 3.13. Нека Σ е азбука. Функцията $\mathrm{Suf}\colon \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star)$ дефинираме за $A \in \mathcal{P}(\Sigma^\star)$ така

$$\mathrm{Suf}(A) = \{\beta \in \Sigma^\star \mid (\exists \alpha \in A)(\beta \succeq \alpha)\}.$$

Твърдение 3.9. *Нека $\mathcal{C} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за $P^{(n)}$. Тогава*

$$(\forall \alpha \in Q)(\forall a \in \widetilde{\Sigma} \cup \{\$\})(!\,\delta(\alpha, a) \implies (\forall \beta \in \mathrm{Suf}(\{\alpha\}))(!\,\delta(\beta, a))).$$

Доказателство. Нека $\alpha \in Q$ и $a \in \widetilde{\Sigma}$ са такива, че $!\,\delta(\alpha, a)$. Нека $\beta \in \mathrm{Suf}(\{\alpha\})$. Съгласно дефиницията на δ, щом $!\,\delta(\alpha, a)$, то $\#_\mathcal{C}(\alpha a) > 0$. Тоест съществува $1 \leq i \leq m$, такова че $\#_{\alpha_i}(\alpha a) > 0$. По-конкретно, съществуват $1 \leq i_1 \leq i_2 \leq |\alpha_i|$, такива че $a^i_{i_1} a^i_{i_1+1} \dots a^i_{i_2} = \alpha a$, където с a^i_l за $1 \leq l \leq |\alpha_i|$ означаваме l-тата дума в изречението α_i. Но тогава и $\#_\mathcal{C}(\beta a) > 0$, тъй като $a^i_{i_2-|\beta|} a^i_{i_2-|\beta|+1} \dots a^i_{i_2} = \beta a$, тоест $\beta \in Q$ и $!\,\delta(\beta, a)$. $\square$

От предходното твърдение следва, че ако дадено състояние α на f-преобразувател на ниво думи за изгладен n-грамен езиков модел има δ-преход с дума a от речника, то всяко, достижимо от α с f-преходи, състояние също ще има δ-преход с тази дума.

3.3 Псевдокод за конструкцията на f-преобразувател на ниво думи

Псевдокода за конструкцията от определение 3.12 ще представим, подобно на псевдокода за конструкцията от определение 2.7, под формата на три функции WordFTransducerStates,

WordFTransducerTransitions и WordFTransducer. Тук също е в сила, че финалните състояния на f-преобразувателите от определение 3.12 са еквивалентни и могат да бъдат заменени с едно единствено финално състояние, което ще отбелязваме с \$.

В алгоритъм 3.1 е описана функцията WordFTransducerStates, която намира състоянията и f-преходите на f-преобразувателя на ниво думи за изгладения n-грамен езиков модел от ARPA описанието на езиковия модел. Тъй като знаем, че състоянията на f-преобразувателя от определение 3.12 са същите като състоянията на преобразувателя от определение 2.7, функцията WordFTransducerStates е същата като WordTransducerStates от алгоритъм 2.2 с разликата, че за всяко състояние α се пазят стойностите $f(\alpha)$ и $\varphi(\alpha)$. Поради тази причина е ясно, че времевата сложност на WordFTransducerStates е $O(n^2 \sum_{i=1}^{m} |\alpha_i|)$, където $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпусът, от който е научен езиковия модел, представен чрез структурата от данни P.

Алгоритъм 3.1. Намира състоянията и f-преходите на f-преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1:  function WordFTransducerStates(P)
2:      Q ← NewHashTable()
3:      HashInsert(Q, ε, 1)
4:      HashInsert(Q, $, 2)
5:      if |P| > 1 then
6:          for all (d_α, α, e_α) ∈ P[2] do
7:              if HashSearch(Q, α[1]) = NIL then
8:                  HashInsert(Q, α[1], |Q| + 1)
9:      E ← NewArray(|Q|, 0)
10:     f ← NewArray(|Q|, 0)
11:     φ ← NewArray(|Q|, 0)
12:     for all (d_α, α, e_α) ∈ P[1] do
13:         state = HashSearch(Q, α)
14:         if state ≠ NIL then
15:             E[state] = e_α
16:     for i ← 2, |P| − 1 do
17:         for all (d_α, α, e_α) ∈ P[i] do
18:             if α[i] ≠ $ then
19:                 HashInsert(Q, α, |Q| + 1)
20:                 ArrayAppend(E, e_α)
21:                 fail ← HashSearch(Q, α[2 : i])
22:                 ArrayAppend(f, fail)
23:                 ArrayAppend(φ, E[fail])
24:     return (Q, f, φ)
```

В алгоритъм 3.2 е описана функцията WordFTransducerTransitions, която намира δ-преходите на f-преобразувателя на ниво думи за изгладен n-грамен езиков модел по дадени – ARPA описанието на езиковия модел и състоянията (заедно с техните f-преходи) на f-преобразувателя. Конструкцията гласи, че f-преобразувателят има δ-преход от всяко състояние α с дума a и изход $d_{\alpha a}$ в състоянието $g_{\mathcal{C}}(h_n(\alpha a))$ (или \$, в случай че $a = \$$), ако αa се среща в корпуса или $\alpha = \varepsilon$. Точно това е отразено в алгоритъм 3.2. Времевата сложност на функцията WordFTransducerTransitions е $O(n|\delta|)$, защото при всяка итерация на вътрешния **for** цикъл (освен единствената итерация, при която $\alpha = \hat{}$) се добавя нов преход,

а това става за $O(|\alpha|) \subseteq O(n)$ време.

Алгоритъм 3.2. Намира δ-преходите на f-преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

```
1:  function WordFTransducerTransitions(P, Q)
2:      δ ← NewHashTable()
3:      λ ← NewHashTable()
4:      for i ← 1, |P| do
5:          for all (d_α, α, e_α) ∈ P[i] do
6:              if α ≠ ^ then
7:                  source ← HashSearch(Q, α[1 : i − 1])
8:                  target ← 1
9:                  if α = $ then
10:                     target ← 2
11:                 else if HashSearch(Q, History(α, n)) ≠ NIL then
12:                     target ← HashSearch(Q, History(α, n))
13:                 HashInsert(δ, (source, α[i]), target)
14:                 HashInsert(λ, (source, α[i]), d_α)
15:     return (δ, λ)
```

Функцията WordFTransducer в алгоритъм 3.3 имплементира конструкцията от определение 3.12. Резултатът от изпълнението на WordFTransducer е f-преобраувателя на ниво думи за дадения изгладен n-грамен езиков модел. По-конкретно, резултатът се състои от

- s – номерът на началното състояние на f-преобразувателя;
- δ, λ – хеш-таблици, които съпоставят на ключ (α, a) съответно състоянието $\delta(\alpha, a)$ и изхода $\lambda(\alpha, a)$;
- f, φ – масиви с размер $|Q|$, които съпоставят на всяко състояние α съответно състоянието $f(\alpha)$ и изходът $\varphi(\alpha)$.

Резултат NIL при търсене в хеш-таблиците δ и λ или резултат 0 при търсене в масивите f и φ може да се интерпретира като липсващ δ-преход или f-преход. Тъй като финалното състояние е винаги с номер 2, това представяне на f-преобразувателя го определя еднозначно с точност до преименуване на състоянията. Така, позовавайки се на твърдение 3.7, получаваме имплементация на конструкцията с времева сложност $O(n(|\widetilde{\Sigma}| + n\sum_{i=1}^{m}|\alpha_i|))$ и пространствена сложност $O(|\widetilde{\Sigma}| + n\sum_{i=1}^{m}|\alpha_i|)$.

Алгоритъм 3.3. Построява f-преобразувателя на ниво думи за дадения изгладен n-грамен езиков модел.

1: **function** WordFTransducer(P)
2: $(Q, F, f, \varphi) \leftarrow$ WordFTransducerStates(P)
3: $(\delta, \lambda) \leftarrow$ WordFTransducerTransitions(P, Q)
4: $s \leftarrow 1$
5: **if** $|P| > 1$ **then**
6: $s \leftarrow$ HashSearch(Q, ˆ)
7: **return** $(s, \delta, f, \lambda, \varphi)$

4 Представяне на езикови модели чрез f-преобразуватели на ниво символи

Досега гледахме на думите от даден речник като на неделими единици – букви. Оттук насетне ще гледаме на тях като на думи над дадена азбука. Поради тази причина ще въведем функция (моноиден хомоморфизъм), която да съпоставя на всяка дума от даден речник последователност от букви, тоест да казва как се изписва всяка дума от речника.

Определение 4.1. Нека $\mathcal{M} = (M, \circ_M, e_M)$ и $\mathcal{N} = (N, \circ_N, e_N)$ са моноиди. Функцията $\rho\colon M \to N$ наричаме *моноиден хомоморфизъм*, ако

- $\rho(e_M) = e_N$;
- $(\forall a, b \in M)(\rho(a \circ_M b) = \rho(a) \circ_N \rho(b))$.

Ако моноидния хомоморфизъм ρ е биекция, ще казваме, че ρ е *моноиден изоморфизъм*.

За всяка азбука Σ, моноид $(M, \circ_M, e_M)$ и моноиден хомоморфизъм $\rho\colon \Sigma^\star \to M$ е в сила, че ρ се определя еднозначно от действието ѝ върху буквите от Σ. Тоест достатъчно е да знаем как се изписват само думите от речника, за да можем да изписваме цели изречения.

Целта ни ще бъде по даден речник $\widetilde{\Sigma}_1$, корпус $\mathcal{C}$ над $\widetilde{\Sigma}_1$, изгладен n-грамен езиков модел $P^{(n)}$ за $\mathcal{C}$ и моноиден хомоморфизъм $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to \Sigma_2^\star$ да построим f-преобразувател $\mathcal{F}$, който представя $P^{(n)}$ по следния начин

$$(\forall \alpha \in \widetilde{\Sigma}_1^\star)(O_{\mathcal{F}}(\rho(\alpha\$)) = P^{(n)}(\{\hat{}\,\alpha\$\})).$$

Тоест искаме $\mathcal{F}$ да представя езиковия модел $P^{(n)}$ като за вход получава вече изписани чрез ρ изречения. Затова $\mathcal{F}$ трябва да може по дадено изписване $\rho(\alpha)$ на изречението α да определи думите, които съставят α. Поради тази причина въвеждаме нов специален символ $\#$, който ще използваме, за да отбелязваме в $\rho(\alpha)$ границите на изписванията на думите от α. Друго съществено условие, което ще изискваме, за да можем да възстановим α от $\rho(\alpha)$, е ρ да бъде инекция, тоест да можем по записа на дадена дума да възстановим еднозначно самата думата. Функция ρ, която удовлетворява гореописаните условия, ще наричаме разделител.

Определение 4.2. Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки и $\# \notin \Sigma_2$. Функцията $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ наричаме *разделител*, ако ρ е моноиден хомоморфизъм, ρ е инекция и

$$(\forall \sigma \in \widetilde{\Sigma}_1 \cup \{\$\})(\rho(\sigma) \in \Sigma_2^\star \circ \{\#\}).$$

4.1 f-преобразувател на ниво символи за езиков модел

Определение 4.3. Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки и $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Нека $\mathcal{C}$ е корпус над $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$. Тогава f-преобразувателя $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota')$, където

- $Q = \{(\alpha', a') \mid \alpha' \in Q' \wedge a' \in \mathrm{Pref}(\{\rho(a) \mid (\alpha', a) \in \mathrm{Dom}(\delta')\}) \wedge \# \not\geq a'\}$;
- $s = (s', \varepsilon)$;
- $F = F' \times \{\varepsilon\}$;
- за всяко $(\alpha, \beta) \in Q$ и $a \in \Sigma_2 \cup \{\#\}$

$$\delta((\alpha,\beta),a) = \begin{cases} (\alpha, \beta a), & \text{ако } (\alpha,\beta a) \in Q \\ (\delta'(\alpha, \rho^{-1}(\beta a)), \varepsilon), & \text{ако } (\alpha,\beta a) \notin Q, a = \# \text{ и } !\delta'(\alpha,\rho^{-1}(\beta a)) \\ \neg!, & \text{иначе} \end{cases}$$

$$\lambda((\alpha,\beta),a) = \begin{cases} 1, & \text{ако } !\delta((\alpha,\beta),a) \text{ и } a \neq \# \\ \lambda'(\alpha, \rho^{-1}(\beta a)), & \text{ако } !\delta((\alpha,\beta),a) \text{ и } a = \# \\ \neg!, & \text{иначе} \end{cases}$$

- за всяко $(\alpha, \beta) \in Q$

$$f((\alpha,\beta)) = \begin{cases} (f'(\alpha), \beta), & \text{ако } !\, f'(\alpha) \\ \neg!, & \text{иначе} \end{cases} \qquad \varphi((\alpha,\beta)) = \begin{cases} \varphi'(\alpha), & \text{ако } !f((\alpha,\beta)) \\ \neg!, & \text{иначе} \end{cases}$$

наричаме *f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ.*

Когато не се интересуваме от разделителя ρ, ще използваме термина „f-преобразувател на ниво символи за изгладен n-грамен езиков модел $P^{(n)}$". Подобно на f-преобразувателите на ниво думи, f-преобразувателите на ниво символи имат ацикличен f-граф, тоест съответните им level_f функции са дефинирани за всяко състояние.

Твърдение 4.1. *Нека $\mathcal{F} = (\Sigma \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво символи за изгладения n-грамен езиков модел $P^{(n)}$. Тогава f-графът $G_f = (Q, E_f)$ на $\mathcal{F}$ е ацикличен.*

Доказателство. Съгласно дефиницията на f, за ребро $((\alpha,\alpha'),(\beta,\beta')) \in E_f$ в G_f е в сила, че $|\alpha| > |\beta|$ и по-точно, че $|\alpha| = |\beta| + 1$. Ако допуснем, че в G_f има цикъл $(\alpha_1,\alpha_1'),(\alpha_2,\alpha_2'),\ldots,(\alpha_m,\alpha_m'),(\alpha_1,\alpha_1')$, то би следвало, че $|\alpha_1| > |\alpha_1|$, което е абсурд. Следователно G_f е ацикличен. □

Следващото твърдение изразява връзката между f-графите съответно на f-преобразувателя на ниво думи и кой да е f-преобразувател на ниво символи за даден изгладен n-грамен езиков модел.

Твърдение 4.2. *Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво символи за $P^{(n)}$. Тогава*

$$(\forall (\alpha,\beta) \in Q)(\mathrm{level}_f((\alpha,\beta)) = \mathrm{level}_{f'}(\alpha)).$$

Доказателство. Твърдението ще докажем с индукция по $\mathrm{level}_f((\alpha,\beta))$.

$\mathbf{level_f((\alpha,\beta)) = 0}$
Тогава $\neg !\, f((\alpha,\beta))$ и следователно $\neg !\, f'(\alpha)$, тоест $\mathrm{level}_{f'}(\alpha) = 0$.

$\mathbf{level_f((\alpha,\beta)) \rightsquigarrow level_f((\alpha,\beta)) + 1}$
Нека $(\alpha,\beta) \in Q$, $\mathrm{level}_f((\alpha,\beta)) > 0$ и твърдението е в сила за всяко състояние $(\alpha',\beta') \in Q$, такова че $\mathrm{level}_f((\alpha',\beta')) = \mathrm{level}_f((\alpha,\beta)) - 1$. Щом $\mathrm{level}_f((\alpha,\beta)) > 0$, то $!\, f((\alpha,\beta))$, което означава, че $!\, f'(\alpha)$ и

$$\begin{aligned}\mathrm{level}_f((\alpha,\beta)) &= \mathrm{level}_f(f((\alpha,\beta))) + 1\\ &= \mathrm{level}_f((f'(\alpha),\beta)) + 1\\ &= \mathrm{level}_{f'}(f'(\alpha)) + 1\\ &= \mathrm{level}_{f'}(\alpha).\end{aligned}$$

□

За да докажем, че f-преобразувателите от определение 4.3 представят езиковите модели, за които са конструирани, ще покажем как траверсирането им съответства на траверсирането на f-преобразувателите от определение 3.12 за същите езикови модели. В тази връзка ще дефинираме няколко помощни понятия и ще разгледаме някои техни свойства.

Определение 4.4. Нека $(\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Дефинираме функциите $S, \overline{S}\colon Q \times (\Sigma_2 \cup \{\#\})^\star \to \mathcal{P}(Q)$ за $\alpha \in Q$ и $\beta \in (\Sigma_2 \cup \{\#\})^\star$ така

$$\begin{aligned}S(\alpha,\beta) &= \{\alpha' \in \mathrm{Suf}(\{\alpha\}) \mid (\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a) \wedge !\, \delta(\alpha', a))\},\\ \overline{S}(\alpha,\beta) &= \mathrm{Suf}(\{\alpha\}) \setminus S(\alpha,\beta).\end{aligned}$$

Твърдение 4.3. *Нека $(\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Тогава за всяко $\alpha \in Q$ и всяко $\beta \in (\Sigma_2 \cup \{\#\})^\star$, за които*

$$\neg(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a) \wedge !\, \delta(\alpha, a)),$$

е в сила, че

- $S(\alpha,\beta) = S(f(\alpha),\beta)$;
- $\overline{S}(\alpha,\beta) = \overline{S}(f(\alpha),\beta) \cup \{\alpha\}$.

Доказателство. Нека $\alpha' \in S(\alpha,\beta)$. Тогава $\alpha' \in \mathrm{Suf}(\{\alpha\})$ и съществува $a \in \widetilde{\Sigma}_1 \cup \{\$\}$, такова че $\beta \preceq \rho(a)$ и $!\, \delta(\alpha', a)$. Съгласно условието на твърдението, $\alpha' \neq \alpha$. Следователно $\alpha' \in \mathrm{Suf}(\{f(\alpha)\})$ и $\alpha' \in S(f(\alpha),\beta)$. Обратно, нека $\alpha' \in S(f(\alpha),\beta)$. Тогава $\alpha' \in \mathrm{Suf}(\{f(\alpha)\})$ и съществува $a \in \widetilde{\Sigma}_1 \cup \{\$\}$, такова че $\beta \preceq \rho(a)$ и $!\, \delta(\alpha', a)$. Но $\mathrm{Suf}(\{f(\alpha)\}) \subseteq \mathrm{Suf}(\{\alpha\})$. Следователно $\alpha' \in \mathrm{Suf}(\{\alpha\})$ и $\alpha' \in S(\alpha,\beta)$. Така показахме, че

$$S(\alpha,\beta) = S(f(\alpha),\beta).$$

Сега втората част от твърдението следва директно

$$\begin{aligned}\overline{S}(\alpha,\beta) &= \mathrm{Suf}(\{\alpha\}) \setminus S(\alpha,\beta)\\ &= (\mathrm{Suf}(\{f(\alpha)\}) \cup \{\alpha\}) \setminus S(f(\alpha),\beta)\\ &= (\mathrm{Suf}(\{f(\alpha)\}) \setminus S(f(\alpha),\beta)) \cup \{\alpha\}\\ &= \overline{S}(f(\alpha),\beta) \cup \{\alpha\}.\end{aligned}$$

□

Следващото твърдение е директно следствие от свойството на f-преобразувателите на ниво думи за изгладени n-грамни езикови модели, формулирано в твърдение 3.9.

Твърдение 4.4. *Нека $(\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Тогава за всяко $\alpha \in Q$, $\beta \in (\Sigma_2 \cup \{\#\})^\star$ и $\gamma \in \mathrm{Suf}(\{\alpha\})$*

$$\gamma \in S(\alpha, \beta) \implies \mathrm{Suf}(\{\gamma\}) \subseteq S(\alpha, \beta). \qquad \square$$

Сега ще опишем поведението на f-преобразувателите от определение 4.3 при прочитане на един входен символ.

Твърдение 4.5. *Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава за всяко $(\alpha, \beta) \in Q \setminus F$ и за всяко $b \in \Sigma_2 \cup \{\#\}$ е в сила, че*

$$(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta b \preceq \rho(a)) \implies \begin{cases} \delta_f((\alpha,\beta),b) & = \begin{cases} (\mathrm{argmax}_{\alpha' \in S(\alpha,\beta b)} |\alpha'|, \beta b), & \text{ако } b \neq \# \\ (\delta'_{f'}(\alpha, \rho^{-1}(\beta b)), \varepsilon), & \text{ако } b = \# \end{cases} \\ \lambda_f((\alpha,\beta),b) & = \begin{cases} \prod_{\alpha' \in \overline{S}(\alpha,\beta b)} e_{\alpha'}, & \text{ако } b \neq \# \\ \lambda'_{f'}(\alpha, \rho^{-1}(\beta b)), & \text{ако } b = \# \end{cases} \end{cases}$$

Доказателство. Твърдението ще докажем с индукция по $\mathrm{level}_f((\alpha, \beta))$.

$\mathrm{level}_f((\alpha, \beta)) = 0$

Щом $\mathrm{level}_f((\alpha, \beta)) = 0$, то $!\,\delta((\alpha, \beta), b)$ и от дефинициите на δ и λ следва, че

$$\begin{aligned} \delta_f((\alpha,\beta),b) = \delta((\alpha,\beta),b) &= \begin{cases} (\alpha, \beta b), & \text{ако } b \neq \# \\ (\delta'(\alpha, \rho^{-1}(\beta b)), \varepsilon), & \text{ако } b = \# \end{cases} \\ &= \begin{cases} (\mathrm{argmax}_{\alpha' \in S(\alpha,\beta b)} |\alpha'|, \beta b), & \text{ако } b \neq \# \\ (\delta'_{f'}(\alpha, \rho^{-1}(\beta b)), \varepsilon), & \text{ако } b = \# \end{cases} \\ \lambda_f((\alpha,\beta),b) = \lambda((\alpha,\beta),b) &= \begin{cases} 1, & \text{ако } b \neq \# \\ \lambda'(\alpha, \rho^{-1}(\beta b)), & \text{ако } b = \# \end{cases} \\ &= \begin{cases} \prod_{\alpha' \in \overline{S}(\alpha,\beta b)} e_{\alpha'}, & \text{ако } b \neq \# \\ \lambda'_{f'}(\alpha, \rho^{-1}(\beta b)), & \text{ако } b = \# \end{cases} \end{aligned}$$

При $b \neq \#$ от дефиницията на δ знаем, че в този случай $(\alpha, \beta b) \in Q$, което означава, че

$$(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta b \preceq \rho(a) \wedge !\,\delta'(\alpha, a)).$$

Тоест $\alpha \in S(\alpha, \beta b)$ и следователно $\alpha = \mathrm{argmax}_{\alpha' \in S(\alpha,\beta b)} |\alpha'|$. Щом $\alpha \in S(\alpha, \beta b)$, то $S(\alpha, \beta b) = \mathrm{Suf}(\{\alpha\})$, $\overline{S}(\alpha, \beta b) = \varnothing$ и $\prod_{\alpha' \in \overline{S}(\alpha,\beta b)} e_{\alpha'} = 1$.

При $b = \#$ от дефиницията на δ знаем, че в този случай $!\,\delta'(\alpha, \rho^{-1}(\beta b))$, което означава, че $\delta'_{f'}(\alpha, \rho^{-1}(\beta b)) = \delta'(\alpha, \rho^{-1}(\beta b))$ и $\lambda'_{f'}(\alpha, \rho^{-1}(\beta b)) = \lambda'(\alpha, \rho^{-1}(\beta b))$.

$\mathrm{level}_f((\alpha, \beta)) \rightsquigarrow \mathrm{level}_f((\alpha, \beta)) + 1$

Нека $(\alpha, \beta) \in Q$, $\mathrm{level}_f((\alpha, \beta)) > 0$ и твърдението е в сила за всяко състояние $(\alpha', \beta') \in Q$, за което $\mathrm{level}_f((\alpha', \beta')) = \mathrm{level}_f((\alpha, \beta)) - 1$.

1сл. $!\delta((\alpha,\beta),b)$

Тогава можем да повторим разсъжденията от базовия случай.

2сл. $\neg!\delta((\alpha,\beta),b)$

От дефинициите на δ_f и λ_f и индукционното предположение следва, че

$$\begin{aligned}
\delta_f((\alpha,\beta),b) &= \delta_f(f((\alpha,\beta)),b) = \delta_f((f'(\alpha),\beta),b)\\
&= \begin{cases} (\operatorname{argmax}_{\alpha'\in S(f'(\alpha),\beta b)} |\alpha'|, \beta b), & \text{ако } b \neq \# \\ (\delta'_{f'}(f'(\alpha), \rho^{-1}(\beta b)), \varepsilon), & \text{ако } b = \# \end{cases}\\
&= \begin{cases} (\operatorname{argmax}_{\alpha'\in S(\alpha,\beta b)} |\alpha'|, \beta b), & \text{ако } b \neq \# \\ (\delta'_{f'}(\alpha, \rho^{-1}(\beta b)), \varepsilon), & \text{ако } b = \# \end{cases}\\
\lambda_f((\alpha,\beta),b) &= \varphi((\alpha,\beta)) \cdot \lambda_f(f((\alpha,\beta)),b) = \varphi'(\alpha)\cdot\lambda_f((f'(\alpha),\beta),b)\\
&= \begin{cases} e_\alpha \cdot \prod_{\alpha'\in\overline{S}(f'(\alpha),\beta b)} e_{\alpha'}, & \text{ако } b \neq \# \\ e_\alpha \cdot \lambda'_{f'}(f'(\alpha), \rho^{-1}(\beta b)), & \text{ако } b = \# \end{cases}\\
&= \begin{cases} \prod_{\alpha'\in\overline{S}(\alpha,\beta b)} e_{\alpha'}, & \text{ако } b \neq \# \\ \lambda'_{f'}(\alpha, \rho^{-1}(\beta b)), & \text{ако } b = \# \end{cases}
\end{aligned}$$

При $b \neq \#$ щом $\neg\,!\,\delta((\alpha,\beta),b)$, то

$$\neg(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta b \preceq \rho(a) \wedge\, !\,\delta'(\alpha,a)).$$

Следователно $\alpha \notin S(\alpha,\beta b)$, $S(\alpha,\beta b) = S(f'(\alpha),\beta b)$, $\overline{S}(\alpha,\beta b) = \overline{S}(f'(\alpha),\beta b)\cup\{\alpha\}$ и

$$\begin{aligned}
\operatorname*{argmax}_{\alpha'\in S(f'(\alpha),\beta b)} |\alpha'| &= \operatorname*{argmax}_{\alpha'\in S(\alpha,\beta b)} |\alpha'|,\\
e_\alpha \cdot \prod_{\alpha'\in\overline{S}(f'(\alpha),\beta b)} e_{\alpha'} &= \prod_{\alpha'\in\overline{S}(\alpha,\beta b)} e_{\alpha'}.
\end{aligned}$$

При $b = \#$ щом $\neg\,!\,\delta((\alpha,\beta),b)$, то $\neg\,!\,\delta'(\alpha,\rho^{-1}(\beta b))$. От предпоставката на твърдението знаем, че $(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\rho(a) = \beta b)$. Съгласно твърдение 3.6, $(\alpha,a) \in \operatorname{Dom}(\delta'_{f'}) = \operatorname{Dom}(\lambda'_{f'})$. Следователно

$$\begin{aligned}
\delta'_{f'}(\alpha,\rho^{-1}(\beta b)) &= \delta'_{f'}(f'(\alpha),\rho^{-1}(\beta b)),\\
\lambda'_{f'}(\alpha,\rho^{-1}(\beta b)) &= e_\alpha \cdot \lambda'_{f'}(f'(\alpha),\rho^{-1}(\beta b)).
\end{aligned}$$

□

В следващите две твърдения ще разгледаме две свойства на функциите S и $\overline{S}$, които ще използваме в доказателството на твърдението, описващо поведението на f-преобразувател на ниво символи за изгладен n-грамен езиков модел при прочитане на последователност от входни символи.

Твърдение 4.6. *Нека $(\widetilde{\Sigma}_1\cup\{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\rho\colon (\widetilde{\Sigma}_1\cup\{\$\})^\star \to (\Sigma_2\cup\{\#\})^\star$ е разделител. Тогава за всяко $\alpha \in Q$, $\beta \in (\Sigma_2\cup\{\#\})^\star$ и $\beta' \preceq \beta$ е в сила, че*

- $\operatorname{argmax}_{\alpha''\in S(\operatorname{argmax}_{\alpha'\in S(\alpha,\beta')}|\alpha'|,\beta)} |\alpha''| = \operatorname{argmax}_{\alpha'\in S(\alpha,\beta)} |\alpha'|$;
- $\overline{S}(\alpha,\beta') \cup \overline{S}(\operatorname{argmax}_{\alpha'\in S(\alpha,\beta')}|\alpha'|,\beta) = \overline{S}(\alpha,\beta)$.

Доказателство.

- Нека $\gamma = \operatorname{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|$, тоест

$$\gamma \succeq \alpha \wedge (\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta' \preceq \rho(a) \wedge ! \, \delta(\gamma, a)).$$

  Достатъчно е да покажем, че $S(\gamma, \beta) = S(\alpha, \beta)$.

  Нека $\gamma' \in S(\gamma, \beta)$. Тогава е в сила, че

$$\gamma' \succeq \gamma \succeq \alpha \wedge (\exists a' \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a') \wedge ! \, \delta(\gamma', a')).$$

  Но това означава, че $\gamma' \in S(\alpha, \beta)$.

  Нека сега $\gamma' \in S(\alpha, \beta)$. Тогава е вярно, че

$$\gamma' \succeq \alpha \wedge (\exists a' \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta' \preceq \beta \preceq \rho(a') \wedge ! \, \delta(\gamma', a')).$$

  Тоест $\gamma' \in S(\alpha, \beta')$ и от максималността на γ следва, че $\gamma' \succeq \gamma$. Така получаваме, че $\gamma' \in S(\gamma, \beta)$.

- Използвайки първата част от доказателството и това, че

$$S(\alpha, \beta') = \operatorname{Suf}(\{ \operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'| \}),$$

  можем да заключим, че

$$\begin{aligned}
&\overline{S}(\alpha, \beta') \cup \overline{S}(\operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \beta) \\
&= (\operatorname{Suf}(\{\alpha\}) \setminus S(\alpha, \beta')) \cup (\operatorname{Suf}(\{ \operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'| \}) \setminus S(\operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \beta)) \\
&= (\operatorname{Suf}(\{\alpha\}) \setminus S(\alpha, \beta')) \cup (S(\alpha, \beta') \setminus S(\alpha, \beta)) \\
&= \operatorname{Suf}(\{\alpha\}) \setminus S(\alpha, \beta) \\
&= \overline{S}(\alpha, \beta).
\end{aligned}$$

□

Твърдение 4.7. *Нека $(\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Тогава за всяко $\alpha \in Q$ и всяко $a \in \widetilde{\Sigma}_1 \cup \{\$\}$, такова че $\rho(a) = \beta' b$, е в сила, че*

- $\delta_f(\alpha, a) = \delta_f(\operatorname{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, a)$;
- $\lambda_f(\alpha, a) = \left(\prod_{\alpha' \in \overline{S}(\alpha,\beta')} e_{\alpha'} \right) \cdot \lambda_f(\operatorname{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, a)$.

Доказателство. Твърдението ще докажем с индукция по $\operatorname{level}_f(\alpha)$.

$\operatorname{level}_f(\alpha) = 0$
Тогава $! \, \delta(\alpha, a)$ и $\alpha \in S(\alpha, \beta')$. Следователно $\alpha = \operatorname{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|$, $S(\alpha, \beta') = \operatorname{Suf}(\{\alpha\})$, $\overline{S}(\alpha, \beta') = \varnothing$ и твърдението е изпълнено.

$\operatorname{level}_f(\alpha) \rightsquigarrow \operatorname{level}_f(\alpha) + 1$
Нека $\alpha \in Q$, $\operatorname{level}_f(\alpha) > 0$ и твърдението е в сила за всяко състояние $\gamma \in Q$, такова че

$\text{level}_f(\gamma) = \text{level}_f(\alpha) - 1$. Ако $!\,\delta(\alpha, a)$, можем да повторим разсъждението от базовия случай. Ако $\neg\,!\,\delta(\alpha, a)$, използвайки индукционното предположение, получаваме, че

$$\begin{aligned}
\delta_f(\alpha, a) &= \delta_f(f(\alpha), a) = \delta_f(\operatorname*{argmax}_{\alpha' \in S(f(\alpha),\beta')} |\alpha'|, a) = \delta_f(\operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, a),\\
\lambda_f(\alpha, a) &= \varphi(\alpha) \cdot \lambda_f(f(\alpha), a)\\
&= e_\alpha \cdot \left(\prod_{\alpha' \in \overline{S}(f(\alpha),\beta')} e_{\alpha'} \right) \cdot \lambda_f(\operatorname*{argmax}_{\alpha' \in S(f(\alpha),\beta')} |\alpha'|, a)\\
&= \left(\prod_{\alpha' \in \overline{S}(\alpha,\beta')} e_{\alpha'} \right) \cdot \lambda_f(\operatorname*{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, a). \qquad \square
\end{aligned}$$

Твърдение 4.8. *Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава за всяко $\alpha \in Q' \setminus F'$ и всяко $\beta \in (\Sigma_2 \cup \{\#\})^\star$*

$$(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a)) \implies \begin{cases} \delta_f^\star((\alpha, \varepsilon), \beta) & = \begin{cases} (\text{argmax}_{\alpha' \in S(\alpha,\beta)} |\alpha'|, \beta), & \text{ако } \# \not\succeq \beta \\ (\delta'_{f'}(\alpha, \rho^{-1}(\beta)), \varepsilon), & \text{ако } \# \succeq \beta \end{cases} \\ \lambda_f^\star((\alpha, \varepsilon), \beta) & = \begin{cases} \prod_{\alpha' \in \overline{S}(\alpha,\beta)} e_{\alpha'}, & \text{ако } \# \not\succeq \beta \\ \lambda'_{f'}(\alpha, \rho^{-1}(\beta)), & \text{ако } \# \succeq \beta \end{cases} \end{cases}$$

Доказателство. Твърдението ще докажем с индукция по $|\beta|$.

$|\beta| = 0$

Тогава $\beta = \varepsilon$ и

$$\begin{aligned}
\delta_f^\star((\alpha, \varepsilon), \beta) &= \delta_f^\star((\alpha, \varepsilon), \varepsilon) = (\alpha, \varepsilon) = (\text{argmax}_{\alpha' \in S(\alpha,\beta)} |\alpha'|, \beta),\\
\lambda_f^\star((\alpha, \varepsilon), \beta) &= \lambda_f^\star((\alpha, \varepsilon), \varepsilon) = 1 = \prod_{\alpha' \in \overline{S}(\alpha,\beta)} e_{\alpha'}.
\end{aligned}$$

Всяко нефинално състояние $\alpha \in Q' \setminus F'$ има изходящ преход в $\mathcal{F}'$ с буква $a \in \widetilde{\Sigma}_1 \cup \{\$\}$. Също така $\beta \preceq \rho(a)$ за всяко $a \in \widetilde{\Sigma}_1 \cup \{\$\}$. Следователно $\alpha = \text{argmax}_{\alpha' \in S(\alpha,\beta)} |\alpha'|$. Това означава, че $S(\alpha, \beta) = \text{Suf}(\{\alpha\})$, $\overline{S}(\alpha, \beta) = \varnothing$ и $\prod_{\alpha' \in \overline{S}(\alpha,\beta)} e_{\alpha'} = 1$.

$|\beta| \rightsquigarrow |\beta| + 1$

Нека $\beta = \beta' b$ и твърдението е в сила за всяка дума с дължина $|\beta'|$. За β' е ясно, че $\# \not\succeq \beta'$, тъй като $\beta \preceq \rho(a)$ и $\rho(a) \in \Sigma_2^\star \circ \{\#\}$ за някое $a \in \widetilde{\Sigma}_1 \cup \{\$\}$. Тогава

$$\begin{aligned}
\delta_f^\star((\alpha, \varepsilon), \beta) &= \delta_f^\star((\alpha, \varepsilon), \beta' b) = \delta_f(\delta_f^\star((\alpha, \varepsilon), \beta'), b) = \delta_f((\text{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \beta'), b)\\
&= \begin{cases} (\text{argmax}_{\alpha'' \in S(\text{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|,\beta)} |\alpha''|, \beta), & \text{ако } b \neq \# \\ (\delta'_{f'}(\text{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \rho^{-1}(\beta)), \varepsilon), & \text{ако } b = \# \end{cases}\\
&= \begin{cases} (\text{argmax}_{\alpha' \in S(\alpha,\beta)} |\alpha'|, \beta) & \text{ако } b \neq \# \\ (\delta'_{f'}(\alpha, \rho^{-1}(\beta)), \varepsilon), & \text{ако } b = \# \end{cases}\\
\lambda_f^\star((\alpha, \varepsilon), \beta) &= \lambda_f^\star((\alpha, \varepsilon), \beta' b) = \lambda_f^\star((\alpha, \varepsilon), \beta') \cdot \lambda_f(\delta_f^\star((\alpha, \varepsilon), \beta'), b)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\prod\nolimits_{\alpha' \in \overline{S}(\alpha,\beta')} e_{\alpha'}\right) \cdot \lambda_f((\mathrm{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \beta'), b) \\
&= \begin{cases} \left(\prod_{\alpha' \in \overline{S}(\alpha,\beta')} e_{\alpha'}\right) \cdot \left(\prod_{\alpha'' \in \overline{S}(\mathrm{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \beta)} e_{\alpha''}\right), & \text{ако } b \neq \# \\ \left(\prod_{\alpha' \in \overline{S}(\alpha,\beta')} e_{\alpha'}\right) \cdot \lambda'_{f'}(\mathrm{argmax}_{\alpha' \in S(\alpha,\beta')} |\alpha'|, \rho^{-1}(\beta)), & \text{ако } b = \# \end{cases} \\
&= \begin{cases} \prod_{\alpha' \in \overline{S}(\alpha,\beta)} e_{\alpha'}, & \text{ако } b \neq \# \\ \lambda'_{f'}(\alpha, \rho^{-1}(\beta)), & \text{ако } b = \# \end{cases}
\end{aligned}
$$

Равенствата следват директно от индукционното предположение и предходните три твърдения. □

От предходното твърдение знаем как се държи един f-преобразувател на ниво символи за изгладен n-грамен езиков модел при прочитане на изписването на една дума от речника. Лесно с индукция по дължината на изречението следва, че f-преобразувателят на ниво символи би се държал по същия начин като f-преобразувателя на ниво думи за същия езиков модел.

Твърдение 4.9. *Нека* $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ *е* f*-преобразувателят на ниво думи за изгладения* n*-грамен езиков модел* $P^{(n)}$ *и* $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ *е* f*-преобразувателят на ниво символи за* $P^{(n)}$ *с разделител* ρ*. Тогава*

$$(\forall \alpha \in \widetilde{\Sigma}_1^\star \circ \{\$\})(\delta_f^\star(s, \rho(\alpha)) = (\delta'^\star_{f'}(s', \alpha), \varepsilon) \wedge \lambda_f^\star(s, \rho(\alpha)) = \lambda'^\star_{f'}(s', \alpha)).$$

Доказателство. Твърдението ще докажем с индукция по $|\alpha|$.

$|\alpha| = 1$

Тогава $\alpha = \$, \# \succeq \rho(\alpha)$ и съгласно предходното твърдение,

$$
\begin{aligned}
\delta_f^\star(s, \rho(\alpha)) &= \delta_f^\star((s', \varepsilon), \rho(\alpha)) = (\delta'_{f'}(s', \alpha), \varepsilon) = (\delta'^\star_{f'}(s', \alpha), \varepsilon), \\
\lambda_f^\star(s, \rho(\alpha)) &= \lambda_f^\star((s', \varepsilon), \rho(\alpha)) = \lambda'_{f'}(s', \alpha) = \lambda'^\star_{f'}(s', \alpha).
\end{aligned}
$$

$|\alpha| \rightsquigarrow |\alpha| + 1$

Нека $\alpha = \alpha' a$ и твърдението е вярно за всяка дума с дължина $|\alpha'|$. Тогава съгласно индукционното предположение и предходното твърдение

$$
\begin{aligned}
\delta_f^\star(s, \rho(\alpha' a)) &= \delta_f^\star((s', \varepsilon), \rho(\alpha')\rho(a)) \\
&= \delta_f^\star(\delta_f^\star((s', \varepsilon), \rho(\alpha')), \rho(a)) \\
&= \delta_f^\star((\delta'^\star_{f'}(s', \alpha'), \varepsilon), \rho(a)) \\
&= (\delta'_{f'}(\delta'^\star_{f'}(s', \alpha'), a), \varepsilon) \\
&= (\delta'^\star_{f'}(s', \alpha), \varepsilon), \\
\lambda_f^\star(s, \rho(\alpha' a)) &= \lambda_f^\star((s', \varepsilon), \rho(\alpha')\rho(a)) \\
&= \lambda_f^\star((s', \varepsilon), \rho(\alpha')) \cdot \lambda_f^\star(\delta_f^\star((s', \varepsilon), \rho(\alpha')), \rho(a)) \\
&= \lambda'^\star_{f'}(s', \alpha') \cdot \lambda_f^\star((\delta'^\star_{f'}(s', \alpha'), \varepsilon), \rho(a)) \\
&= \lambda'^\star_{f'}(s', \alpha') \cdot \lambda'_{f'}(\delta'^\star_{f'}(s', \alpha'), a) \\
&= \lambda'^\star_{f'}(s', \alpha).
\end{aligned}
$$

□

Сега, знаейки че f-преобразувателите от определение 4.3 се държат като f-преобразувателите от определение 3.12 за същите изгладени n-грамни езикови модели, можем да заключим, че f-преобразувателите от определение 4.3 наистина представят съответните им изгладени n-грамни езикови модели.

Твърдение 4.10. *Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава*

- $\mathrm{Dom}(O_{\mathcal{F}}) = \{\rho(\alpha) \mid \widetilde{\Sigma}_1^\star \circ \{\$\}\}$;
- $(\forall \alpha \in \widetilde{\Sigma}_1^\star)(O_{\mathcal{F}}(\rho(\alpha\$)) = P^{(n)}(\{\hat{}\,\alpha\$\}))$.

Доказателство. Нека $\alpha \in \widetilde{\Sigma}_1^\star \circ \{\$\}$. Тогава $\alpha \in \mathrm{Dom}(O_{\mathcal{F}'})$ и

$$\alpha \in \mathrm{Dom}(O_{\mathcal{F}'}) \implies \delta'^\star_{f'}(s', \alpha) \in F' \implies \delta_f^\star(s, \rho(\alpha)) \in F \implies \rho(\alpha) \in \mathrm{Dom}(O_{\mathcal{F}}).$$

Нека сега $\alpha \in \mathrm{Dom}(O_{\mathcal{F}})$. Тогава $\delta_f^\star(s, \alpha) \in F$, тоест $\# \succeq \alpha$, тъй като финалните състояния на $\mathcal{F}$ имат входящи преходи само с буква $\#$. Следователно можем да представим α по следния начин

$$\alpha = \alpha_1 \# \alpha_2 \# \ldots \# \alpha_{m+1} \#,$$

където $\alpha_i \in \Sigma_2^\star$ за $1 \leq i \leq m+1$. Да допуснем, че съществува $1 \leq i \leq m+1$, такова че $\alpha_i \# \notin \mathrm{Rng}(\rho)$. Нека си изберем най-малкото такова i. Тогава съществува префикс β на $\alpha_i \#$, такъв че $\neg(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a))$. Нека β е най-късият такъв префикс. Ясно е, че $\beta \neq \varepsilon$, тоест $\beta = \beta' b$ и $(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta' \preceq \rho(a))$. Дотук знаем, че съществуват $\gamma, \gamma' \in Q'$, такива че

- $\delta_f^\star(s, \alpha_1 \# \alpha_2 \# \ldots \alpha_{i-1} \#) = (\gamma, \varepsilon)$;
- $\delta_f^\star(s, \alpha_1 \# \alpha_2 \# \ldots \alpha_{i-1} \# \beta') = \delta_f^\star((\gamma, \varepsilon), \beta') = (\mathrm{argmax}_{\gamma'' \in S(\gamma, \beta')} |\gamma''|, \beta') = (\gamma', \beta')$;

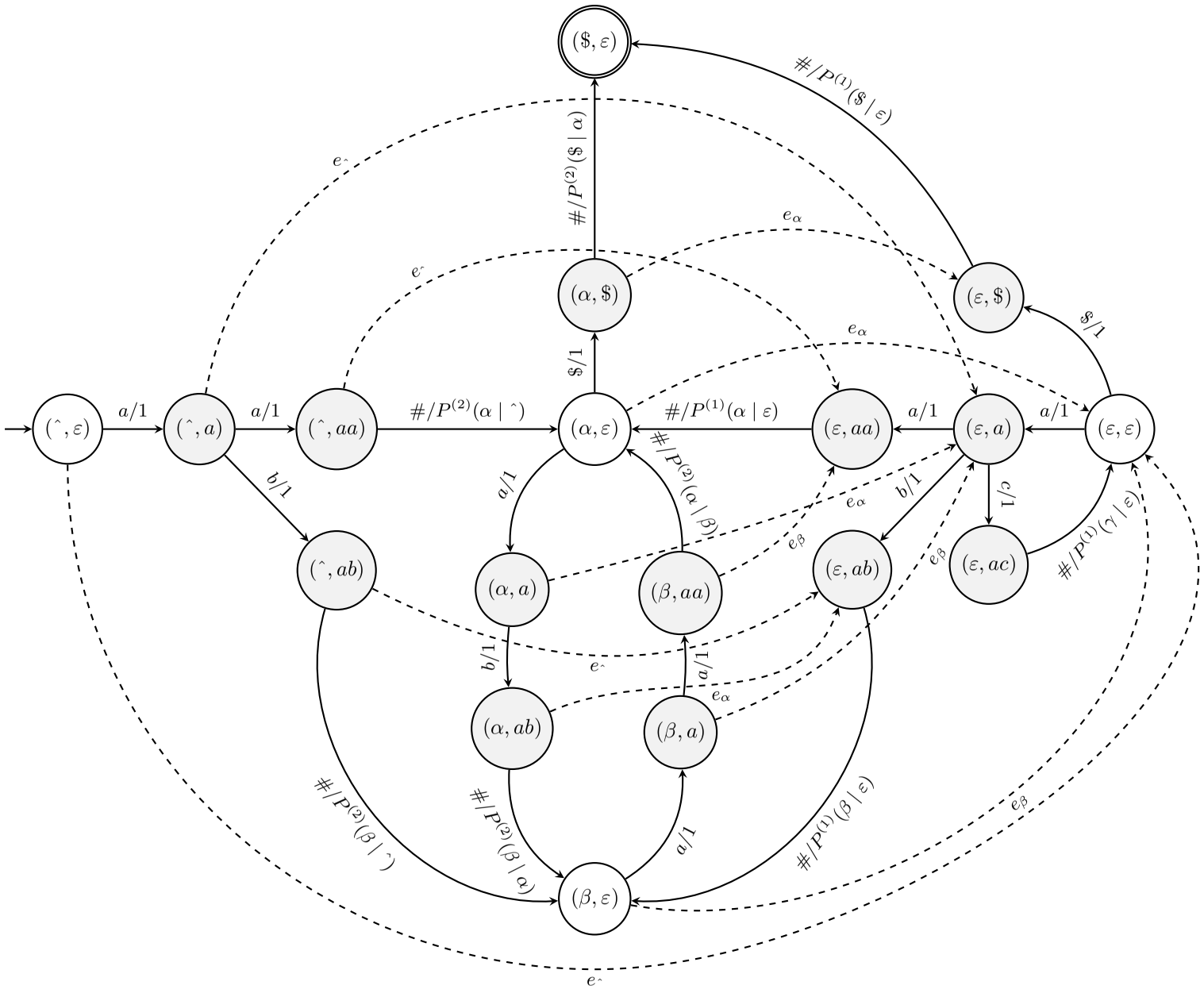
Ясно е, че за всяко $k \in \mathbb{N}$, ако $!\, f^{(k)}((\gamma', \beta'))$, то $f^{(k)}((\gamma', \beta')) = (\gamma'', \beta')$ и $\neg !\, \delta((\gamma'', \beta'), b)$, защото иначе β нямаше да е такова, че $\neg(\exists a \in \widetilde{\Sigma}_1 \cup \{\$\})(\beta \preceq \rho(a))$. Следователно $\neg !\, \delta_f((\gamma', \beta'), b)$, което противоречи на това, че $\alpha \in \mathrm{Dom}(O_{\mathcal{F}})$. Следователно $\alpha_i \# \in \mathrm{Rng}(\rho)$ за всяко $1 \leq i \leq m+1$, което значи, че $\rho^{-1}(\alpha) \in \mathrm{Dom}(\rho) = \widetilde{\Sigma}_1^\star \circ \{\$\}$.

За втората част от твърдението да разгледаме $\alpha \in \widetilde{\Sigma}_1^\star$ и да забележим, че

$$O_{\mathcal{F}}(\rho(\alpha\$)) = \iota \cdot \lambda_f^\star(s, \rho(\alpha\$)) = \iota' \cdot \lambda'^\star_{f'}(s', \alpha\$) = O_{\mathcal{F}'}(\alpha\$) = P^{(n)}(\{\hat{}\,\alpha\$\}). \qquad \square$$

На фигура 4.1 е изобразен f-преобразувателят на ниво символи за $P^{(2)}$ с разделител $\rho = \{(\alpha, aa\#), (\beta, ab\#), (\gamma, ac\#), (\$, \$\#)\}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\,\alpha\beta\alpha\$$, $\hat{}\,\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$. Новите състояния (тези с втора координата различна от ε) спрямо f-преобразувателя на ниво думи за $P^{(2)}$ от фигура 3.2 са защриховани.

Дефинираните в определение 4.3 f-преобразуватели имат размер, зависещ мултилинейно от n и големината на речника (в брой символи) и линейно от големината на корпуса (в брой символи).



Фигура 4.1: f-преобразувателят на ниво символи за $P^{(2)}$ с разделител $\rho = \{(\alpha, aa\#), (\beta, ab\#), (\gamma, ac\#), (\$, \$\#)\}$, където $P^{(2)}$ е изгладен 2-грамен езиков модел за корпуса ($\hat{}\alpha\beta\alpha\$, \hat{}\beta\alpha\$$) над речника $\{\alpha, \beta, \gamma\}$.

Твърдение 4.11. *Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки и $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Нека $\mathcal{C} = (\alpha_1, \alpha_2, \dots \alpha_m)$ е корпус над $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава f-преобразувателят на ниво символи $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ за $P^{(n)}$ с разделител ρ е такъв, че*

- $|Q| \in O(\sum_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} |\rho(a)| + n \sum_{i=1}^{m} |\rho(\alpha_i)|)$;
- $|\operatorname{Dom}(\delta)| \in O(\sum_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} |\rho(a)| + n \sum_{i=1}^{m} |\rho(\alpha_i)|)$.

Доказателство. Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$. Можем да разделим състоянията на $\mathcal{F}$ на такива, съответстващи на състояния на $\mathcal{F}'$, и на нови състояния, съответстващи на разбиването на преходите на $\mathcal{F}'$ на символи.

$$\begin{aligned} Q &= Q_1 \cup Q_2 \\ Q_1 &= \{(\alpha', \varepsilon) \mid \alpha' \in Q'\} \\ Q_2 &= \{(\alpha', a') \mid \alpha' \in Q' \wedge a' \in \operatorname{Pref}(\{\rho(a) \mid (\alpha', a) \in \operatorname{Dom}(\delta')\}) \wedge \# \not\geq a' \wedge a' \neq \varepsilon\} \end{aligned}$$

Аналогично можем да разделим и преходите на $\mathcal{F}$ на такива, влизащи в състояния от Q_1 и съответстващи на преходи от $\mathcal{F}'$, и на нови преходи, влизащи в състояния от Q_2.

$$\begin{aligned} \operatorname{Dom}(\delta) &= \Delta_1 \cup \Delta_2 \\ \Delta_1 &= \{(q, a) \in Q \times (\Sigma_2 \cup \{\#\}) \mid \delta(q, a) \in Q_1\} \\ \Delta_2 &= \{(q, a) \in Q \times (\Sigma_2 \cup \{\#\}) \mid \delta(q, a) \in Q_2\} \end{aligned}$$

Ясно е, че Q_1 има толкова елементи, колкото са състоянията на $\mathcal{F}'$, а от дефиницията на δ следва, че Δ_1 има толкова елементи, колкото са преходите на $\mathcal{F}'$.

Сега ще покажем, че броят състояния в Q_2 съвпада с броя преходи в Δ_2. Да разгледаме функцията $r\colon Q_2 \to \Delta_2$, дефинирана за $(\alpha, \beta b) \in Q_2$ така

$$r((\alpha, \beta b)) = ((\alpha, \beta), b).$$

Ясно е, че r е инекция. Ако $((\alpha, \beta), b) \in \Delta_2$, то $\delta((\alpha, \beta), b) \in Q_2$, тоест $\delta((\alpha, \beta), b) = (\alpha', \beta')$ и от дефиницията на δ знаем, че $\alpha' = \alpha$ и $\beta' = \beta b$. Тоест

$$r((\alpha', \beta')) = r((\alpha, \beta b)) = ((\alpha, \beta), b)$$

и r е сюрекция. Следователно r е биекция и $|Q_2| = |\Delta_2|$.

$$\begin{aligned} |Q_2| &= |\{(\alpha', a') \mid \alpha' \in Q' \wedge a' \in \operatorname{Pref}(\{\rho(a) \mid (\alpha', a) \in \operatorname{Dom}(\delta')\}) \wedge \# \not\geq a' \wedge a' \neq \varepsilon\}| \\ &= |\bigcup_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} \{(\alpha', a') \in Q' \times (\operatorname{Pref}(\{\rho(a)\}) \setminus \{\varepsilon, \rho(a)\}) \mid ! \, \delta'(\alpha', a)\}| \\ &= |\bigcup_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} \{(\alpha', a') \in Q' \times (\operatorname{Pref}(\{\rho(a)\}) \setminus \{\varepsilon, \rho(a)\}) \mid \alpha' = \varepsilon \vee \#_{\mathcal{C}}(\alpha' a) > 0\}| \\ &\leq |Q_2'| + |Q_2''|, \end{aligned}$$

където

$$|Q_2'| = |\bigcup_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} \{\varepsilon\} \times (\operatorname{Pref}(\{\rho(a)\}) \setminus \{\varepsilon, \rho(a)\})| \leq \sum_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} |\rho(a)|,$$

$$
\begin{aligned}
|Q_2''| &= |\bigcup_{a\in\widetilde{\Sigma}_1\cup\{\$\}} \{(\alpha',a')\in Q'\times(\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}) \mid \#_{\mathcal{C}}(\alpha' a)>0\}| \\
&= |\bigcup_{a\in\widetilde{\Sigma}_1\cup\{\$\}} \bigcup_{j=0}^{n-1} \{(\alpha',a')\in Q'\times(\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}) \mid |\alpha'|=j\wedge \#_{\mathcal{C}}(\alpha' a)>0\}| \\
&= |\bigcup_{a\in\Sigma_1\cup\{\$\}} \bigcup_{j=0}^{n-1} \{(\alpha',a')\in Q'\times(\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}) \mid |\alpha'|=j\wedge \sum_{i=1}^{m}\#_{\alpha_i}(\alpha' a)>0\}| \\
&= |\bigcup_{a\in\Sigma_1\cup\{\$\}} \bigcup_{i=1}^{m}\bigcup_{j=0}^{n-1} \{(\alpha',a')\in Q'\times(\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}) \mid |\alpha'|=j\wedge \#_{\alpha_i}(\alpha' a)>0\}| \\
&\le \sum_{j=1}^{m}\sum_{i=1}^{n-1} |\bigcup_{a\in\Sigma_1\cup\{\$\}} \{(\alpha',a')\in Q'\times(\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}) \mid |\alpha'|=j\wedge \#_{\alpha_i}(\alpha' a)>0\}| \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n-1} |\bigcup_{a\in\Sigma_1\cup\{\$\}}\bigcup_{\alpha'\in Q'} \{(\alpha',a') \mid a'\in\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}\wedge|\alpha'|=j\wedge\#_{\alpha_i}(\alpha' a)>0\}| \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n-1} |\bigcup_{\substack{\alpha' a\in(\Sigma_1\cup\{\$\})^{j+1}:\\ \#_{\alpha_i}(\alpha' a)>0}} \{(\alpha',a') \mid a'\in\mathrm{Pref}(\{\rho(a)\})\setminus\{\varepsilon,\rho(a)\}\}| \\
&\le \sum_{i=1}^{m}\sum_{j=1}^{n-1} \sum_{\substack{\alpha' a\in(\Sigma_1\cup\{\$\})^{j+1}:\\ \#_{\alpha_i}(\alpha' a)>0}} |\rho(a)| \\
&\le \sum_{i=1}^{m}\sum_{j=1}^{n-1} |\rho(\alpha_i)| \\
&= n\sum_{i=1}^{m}|\rho(\alpha_i)|.
\end{aligned}
$$

Следователно

$$|Q_2| \le \sum_{a\in\widetilde{\Sigma}_1\cup\{\$\}} |\rho(a)| + n\sum_{i=1}^{m}|\rho(\alpha_i)|.$$

Така за броя състояния и броя преходи на $\mathcal{F}$ получаваме

$$
\begin{aligned}
|Q| = |Q_1| + |Q_2| &\le n\sum_{i=1}^{m}|\alpha_i| + \sum_{a\in\Sigma_1\cup\{\$\}}|\rho(a)| + n\sum_{i=1}^{m}|\rho(\alpha_i)| \\
&\in O\left(\sum_{a\in\Sigma_1\cup\{\$\}}|\rho(a)| + n\sum_{i=1}^{m}|\rho(\alpha_i)|\right); \\
|\,\mathrm{Dom}(\delta)| = |\Delta_1| + |\Delta_2| &\le |\Sigma_1| + n\sum_{i=1}^{m}|\alpha_i| + \sum_{a\in\Sigma_1\cup\{\$\}}|\rho(a)| + n\sum_{i=1}^{m}|\rho(\alpha_i)| \\
&\in O\left(\sum_{a\in\Sigma_1\cup\{\$\}}|\rho(a)| + n\sum_{i=1}^{m}|\rho(\alpha_i)|\right).
\end{aligned}
$$

□

За f-преобразувателите на ниво символи за изгладени n-грамни езикови модели също е вярно, че обработват дадена входна дума за линейно време относно нейната дължина. Нещо повече, броят на δ-преходите и f-преходите, които те правят, са не повече от дължината на входното изречение в брой думи плюс дължината на входното изречение в брой символи.

Определение 4.5. Нека $n \in \mathbb{N}$ и $X_1, X_2, \ldots, X_n$ са множества. Тогава за $1 \leq i \leq n$ и $X \subseteq X_1 \times X_2 \times \ldots \times X_n$ дефинираме

$$\mathrm{Proj}_i(X) = \{x_i \mid (x_1, x_2, \ldots, x_n) \in X\}.$$

Твърдение 4.12. *Нека $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за изгладения n-грамен езиков модел $P^{(n)}$ с разделител $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$. Тогава за всяко $(\alpha, \varepsilon) \in Q \setminus F$ и $a \in \widetilde{\Sigma}_1 \cup \{\$\}$ е в сила, че*

$$\#_\delta((\alpha, \varepsilon), \rho(a)) + \#_f((\alpha, \varepsilon), \rho(a)) \leq |\rho(a)| + |\alpha| - |\,\mathrm{Proj}_1(\delta_f^\star((\alpha, \varepsilon), \rho(a)))| + 1.$$

Доказателство. Първо, с индукция по $\mathrm{level}_f(q)$ ще докажем, че

$$(\forall (q, b) \in \mathrm{Dom}(\delta_f))(\#_f(q, b) \leq |\,\mathrm{Proj}_1(q)| - |\,\mathrm{Proj}_1(\delta_f(q, b))| + \xi_b),$$

където $\xi_b = 1$, ако $b = \#$, и $\xi_b = 0$ в противен случай.

$\mathrm{level}_f(q) = 0$
Тогава $!\,\delta(q, b)$ и $\#_f(q, b) = 0$. Ако $b \neq \#$, то $|\,\mathrm{Proj}_1(q)| = |\,\mathrm{Proj}_1(\delta_f(q, b))|$ и $\xi_b = 0$. Ако $b = \#$, то $|\,\mathrm{Proj}_1(\delta_f(q, b))| \leq |\,\mathrm{Proj}_1(q)| + 1$ (тъй като първата координата на q е състояние от f-преобразувателя на ниво думи за $P^{(n)}$ и за него знаем, че това е вярно) и $\xi_b = 1$. И в двата случая твърдението е в сила.

$\mathrm{level}_f(q) \rightsquigarrow \mathrm{level}_f(q) + 1$
Нека $q \in Q \setminus F$, $\mathrm{level}_f(q) > 0$ и твърдението е в сила за всяко състояние $p \in Q \setminus F$, такова че $\mathrm{level}_f(p) = \mathrm{level}_f(q) - 1$. Нека $b \in \Sigma_2 \cup \{\#\}$.

1. Ако $!\,\delta(q, b)$, разсъждението е същото като в базовия случай.
2. Ако $\neg !\,\delta(q, b)$, то $|\,\mathrm{Proj}_1(q)| = |\,\mathrm{Proj}_1(f(q))| + 1$ (тъй като първата координата на q е състояние от f-преобразувателя на ниво думи за $P^{(n)}$ и за него знаем, че това е вярно) и

$$\begin{aligned}
\#_f(q, b) &= \#_f(f(q), b) + 1 \\
&\leq |\,\mathrm{Proj}_1(f(q))| - |\,\mathrm{Proj}_1(\delta_f(f(q), b))| + \xi_b + 1 \\
&= (|\,\mathrm{Proj}(q)| - 1) - |\,\mathrm{Proj}_1(\delta_f(q, b))| + \xi_b + 1 \\
&= |\,\mathrm{Proj}_1(q)| - |\,\mathrm{Proj}(\delta_f(q, b))| + \xi_b.
\end{aligned}$$

Нека сега $(\alpha, \varepsilon) \in Q \setminus F$, $a \in \widetilde{\Sigma}_1 \cup \{\$\}$ и $\rho(a) = a_1 a_2 \ldots a_{m+1}$. Нека $q_0, q_1, \ldots, q_{m+1}$ е редицата от състояния, за която $q_0 = (\alpha, \varepsilon)$ и $q_i = \delta_f^\star((\alpha, \varepsilon), a_1 a_2 \ldots a_i)$ за $1 \leq i \leq m + 1$. Тогава $\#_\delta((\alpha, \varepsilon), \rho(q)) = |\rho(a)|$ и

$$(\forall 1 \leq i \leq m + 1)(\#_f(q_{i-1}, a_i) \leq |\,\mathrm{Proj}_1(q_{i-1})| - |\,\mathrm{Proj}_1(q_i)| + \xi_{a_i}).$$

Сега, сумирайки по i, получаваме

$$\sum_{i=1}^{m+1} \#_f(q_{i-1}, a_i) \leq \sum_{i=1}^{m+1} (|\,\mathrm{Proj}_1(q_{i-1})| - |\,\mathrm{Proj}_1(q_i)|) + \sum_{i=1}^{m+1} \xi_{a_i}$$

$$\#_f(q_0, \rho(a)) \leq |\operatorname{Proj}_1(q_0)| - |\operatorname{Proj}_1(q_{m+1})| + \xi_{a_{m+1}}$$
$$\#_f((\alpha, \varepsilon), \rho(a)) \leq |\alpha| - |\operatorname{Proj}_1(\delta_f^\star((\alpha, \varepsilon), \rho(a)))| + 1.$$

Тоест общият брой направени преходи е

$$\#_\delta((\alpha, \varepsilon), \rho(a)) + \#_f((\alpha, \varepsilon), \rho(a)) \leq |\rho(a)| + |\alpha| - |\operatorname{Proj}_1(\delta_f^\star((\alpha, \varepsilon), \rho(a)))| + 1. \qquad \square$$

Твърдение 4.13. *Нека $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за изгладения n-грамен езиков модел $P^{(n)}$ с разделител $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$. Тогава за всяко $\alpha \in \widetilde{\Sigma}_1^\star \circ \{\$\}$ е в сила, че*

$$\#_\delta(s, \rho(\alpha)) + \#_f(s, \rho(\alpha)) \leq |\rho(\alpha)| + |\alpha|.$$

Доказателство. Нека $\alpha = a_1 a_2 \dots a_{m+1}$ и $q_0, q_1, \dots, q_{m+1}$ е редицата от състояния, за която $q_0 = s$ и $q_i = \delta_f^\star(s, \rho(a_1 a_2 \dots a_i))$ за $1 \leq i \leq m+1$. От твърдение 4.9 следва, че

$$(\forall 1 \leq i \leq m+1)(\operatorname{Proj}_2(q_i) = \varepsilon),$$

а от твърдение 4.12 следва, че

$$(\forall 1 \leq i \leq m+1)(\#_\delta(q_{i-1}, \rho(a_i)) + \#_f(q_{i-1}, \rho(a_i)) \leq |\rho(a_i)| + |q_{i-1}| - |q_i| + 1).$$

Сумирайки по i, получаваме

$$\begin{aligned}
\sum_{i=1}^{m+1} \#_\delta(q_{i-1}, \rho(a_i)) + \sum_{i=1}^{m+1} \#_f(q_{i-1}, \rho(a_i)) &\leq \sum_{i=1}^{m+1} |\rho(a_i)| + \sum_{i=1}^{m+1} (|q_{i-1}| - |q_i|) + \sum_{i=1}^{m+1} 1 \\
\#_\delta(q_0, \rho(\alpha)) + \#_f(q_0, \rho(\alpha)) &\leq |\rho(\alpha)| + |q_0| - |q_{m+1}| + |\alpha| \\
&\leq |\rho(\alpha)| + |\alpha| + 1 - |q_{m+1}| \\
&\leq |\rho(\alpha)| + |\alpha|. \qquad \square
\end{aligned}$$

f-преобразувателите на ниво символи за изгладени n-грамни езикови модели имат аналогично свойство на това формулирано в твърдение 3.9, а именно, че ако дадено състояние (α, β) на f-преобразувателя има δ-преход с буква a, то всяко състояние (α', β), където α' е суфикс на α, също ще има δ-преход с тази буква.

Твърдение 4.14. *Нека $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за изгладения n-грамен езиков модел $P^{(n)}$ с разделител $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$. Тогава*

$$(\forall (\alpha, \beta) \in Q)(\forall b \in \Sigma_2 \cup \{\#\})(!\,\delta((\alpha, \beta)) \implies (\forall (\alpha', \beta) \in Q)(\alpha' \succeq \alpha \implies !\,\delta((\alpha', \beta'), b))).$$

Доказателство. Нека $\mathcal{F} = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$. Нека $(\alpha, \beta) \in Q$ и $b \in \Sigma_2 \cup \{\#\}$ са такива, че $!\,\delta((\alpha, \beta), b)$. Това означава, че съществува $a \in \widetilde{\Sigma}_1 \cup \{\$\}$, такова че $\beta b \preceq \rho(a) \wedge !\,\delta'(\alpha, a)$. Нека си изберем едно такова a. От твърдение твърдение 3.9 следва, че $!\,\delta'(\alpha', a)$, което означава, че $!\,\delta((\alpha', \beta)), b)$. $\square$

От предходното твърдение следва, че ако дадено състояние q на f-преобразувател на ниво символи за изгладен n-грамен езиков модел има δ-преход с буква a, то всяко достижимо от q с f-преходи, състояние също ще има δ-преход с тази буква.

4.2 Псевдокод за конструкцията на f-преобразувател на ниво символи

Функцията SymbolFTransducer в алгоритъм 4.1 построява f-преобразувателя на ниво символи за изгладен n-грамен езиков модел с разделител ρ от f-преобразувателя на ниво думи за същия езиков модел. Можем да забележим, че псевдокодът следва конструкцията от определение 4.3. Всяка итерация на **for** цикъла от ред 12 добавя преходите във f-преобразувателя на ниво символи, които съответстват на преход (q, a, p) от f-преобразувателя на ниво думи. Тоест за всеки преход (q, a, p) биват добавени преходите

$$(q, a_1, (q_1, a_1)), ((q, a_1), a_2, (q, a_1a_2)), \ldots, ((q, a_1a_2 \ldots a_m), a_{m+1}, p),$$

където $\rho(a) = a_1a_2 \ldots a_{m+1}$. Изходите на всеки от тези преходи е 1 с изключение на последния, чийто изход е изходът на прехода (q, a, p) в f-преобразувателя на ниво думи. Единствената разлика с конструкцията е, че използваме директно състоянието q, а не наредената двойка (q, ε). f-преходите и техните изходи се дефинират в псевдокода по същия начин като в конструкцията от определение 4.3. Тъй като f-графът на изходния f-преобразувател е ацикличен, сортираме топологически неговите върхове и разглеждаме преходите на f-преобразувателя спрямо реда на началните им състояния в сортировката. По този начин сме сигурни, че при разглеждане на преход (q, a, p), състоянията, в които ще влизат новодобавените f-преходи, съществуват. Търсенето в хеш-таблицата δ от ред 16 и условната конструкция от ред 17 осигуряват, че всяко състояние на f-преобразувателя на ниво символи ще бъде добавено само веднъж, тоест няма да има копия, поради преходи във f-преобразувателя на ниво думи от едно и също състояние с думи, чиито изписвания имат общ префикс.

Функцията SymbolFTransducer прави толкова стъпки, колкото е сумата от дължините на изписванията на всички входни думи от преходите на f-преобразувателя на ниво думи, като всяка една такава стъпка се състои от константен брой извиквания на функциите HashInsert и HashSearch. Това означава, че времевата сложност на SymbolFTransducer е $O(\sum_{(q,a,p)\in\delta'} |\rho(a)|)$, считайки че функцията ρ върху входа a отнема линейно време относно размера на изхода $\rho(a)$.

Алгоритъм 4.1. Построява f-преобразувателя на ниво символи за изгладен n-грамен езиков модел с разделител ρ от f-преобразувателя на ниво думи за същия езиков модел.

```
 1: function SymbolFTransducer((s', δ', f', λ', φ'), ρ)
 2:     δ ← NewHash()
 3:     λ ← NewHash()
 4:     f ← f'
 5:     φ ← φ'
 6:     E ← NewArray(|f|, NewList())
 7:     for all (q, a, p) ∈ δ' do
 8:         AppendList(E, (a, p))
 9:     T ← TopologicalSort(f')
10:     for i ← 1, |T| do
11:         q ← T[i]
12:         for all (a, p) ∈ E[q] do
13:             spelling ← ρ(a)
14:             state ← q
15:             for i ← 1, |spelling| − 1 do
16:                 next ← HashSearch(δ, (state, spelling[i]))
17:                 if next = NIL then
18:                     next ← |f| + 1
19:                     HashInsert(δ, (state, spelling[i]), next)
20:                     HashInsert(λ, (state, spelling[i]), 1)
21:                     if f[state] ≠ 0 then
22:                         ArrayAppend(f, HashSearch(δ, (f[state], spelling[i])))
23:                     else
24:                         ArrayAppend(f, 0)
25:                     ArrayAppend(φ, φ'[q])
26:                 state ← next
27:             HashInsert(δ, (state, #), p)
28:             HashInsert(λ, (state, #), HashSearch(λ', (q, a)))
29:     return (s', δ, f, λ, φ)
```

5 Канонизация на преобразуватели и f-преобразуватели

Теорията, представена в Mihov и Schulz, 2018 (in press) и Mohri 2000, гласи, че за да бъде минимизиран един преобразувател, следва той първо да бъде канонизиран. Канонизиран е такъв преобразувател, чиито изходи са изнесени „максимално най-рано", тоест възможно най-близо до началното състояние. В тази глава ще разгледаме задачата за канонизация на f-преобразуватели на ниво думи и ниво символи за изгладени n-грамни езикови модели.

5.1 Канонизация на преобразуватели с изходи от $\mathcal{R}_{[0,1]}$

Първо ще опишем как се канонизират преобразуватели без f-преходи. По-конкретно, ще разгледаме задачата за канонизация на преобразуватели с изходи от $\mathcal{R}_{[0,1]}$. Решението на тази задача е описано в Mohri 1997.

Определение 5.1. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. За $q \in Q$ дефинираме $O^q_{\mathcal{T}} \colon \Sigma^\star \to M$ – *функцията на състоянието* q, за $\alpha \in \Sigma^\star$ така

$$O^q_{\mathcal{T}}(\alpha) = \begin{cases} \lambda^\star(q, \alpha), & \text{ако } \delta^\star(q, \alpha) \in F \\ \neg\,!, & \text{иначе} \end{cases}$$

Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Под максимален изход на състояние $q \in Q$ ще разбираме максимума от изходите на всички пътища в $\mathcal{T}$ от q до финално състояние, тоест

$$\max\{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q, \alpha) \in F\}.$$

За всеки преобразувател с изходи от $\mathcal{R}_{[0,1]}$ и всяко негово състояние, от което има път до финално състояние, този максимум съществува и представлява изхода, акумулиран по някой прост път (път без цикли) от съответното състояние до финално състояние. Горепосоченият максимум ще наричаме максимален изход на състоянието q, ако той съществува и е различен от 0. В противен случай, ще считаме, че максималният изход на q е 1.

Определение 5.2. Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Дефинираме функцията $\mathrm{mso}_{\mathcal{T}} \colon Q \to (0, 1]$ за $q \in Q$ така

$$\mathrm{mso}_{\mathcal{T}}(q) = \begin{cases} \max\{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q, \alpha) \in F\}, & \text{ако } (\exists \alpha \in \Sigma^\star)(\delta^\star(q, \alpha) \in F \wedge O^q_{\mathcal{T}}(\alpha) \neq 0) \\ 1, & \text{иначе} \end{cases}$$

Твърдение 5.1. *Нека* $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ *е преобразувател. Тогава е в сила, че*

$$(\forall q \in F)(\mathrm{mso}_{\mathcal{T}}(q) = 1).$$

Доказателство. Нека $q \in F$. Тогава $\delta^\star(q, \varepsilon) \in F$ и $O^q_{\mathcal{T}}(\varepsilon) = \lambda^\star(q, \varepsilon) = 1$. Следователно

$$\begin{aligned}\mathrm{mso}_{\mathcal{T}}(q) &= \max\{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q, \alpha) \in F\} \\ &= \max(\{O^q_{\mathcal{T}}(\varepsilon)\} \cup \{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^+ \wedge \delta^\star(q, \alpha) \in F\}) \\ &= \max(\{1\} \cup \{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^+ \wedge \delta^\star(q, \alpha) \in F\}) \\ &= 1.\end{aligned}$$

□

Под каноничен преобразувател ще разбираме преобразувател, чиито изходи са от $\mathcal{R}_{[0,1]}$ и са изнесени максимално най-рано, тоест максималните изходи на състоянията са равни на 1.

Определение 5.3. Преобразувател $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ наричаме *каноничен*, ако

$$(\forall q \in Q)(\mathrm{mso}_{\mathcal{T}}(q) = 1).$$

За да канонизираме даден преобразувател с изходи от $\mathcal{R}_{[0,1]}$ ще пренасяме максималните изходи на състоянията от излизащите върху влизащите преходи. За целта ще използваме операцията „деление на реални числа", която е дефинирана по обичайния начин за всяка двойка реални числа от $\mathbb{R} \times (\mathbb{R} \setminus \{0\})$ и която ще означаваме с $\div$.

Определение 5.4. Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. *Канонична форма* на $\mathcal{T}$ наричаме преобразувателя $\mathcal{T}' = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda', \iota')$, където

- $\iota' = \iota \cdot \mathrm{mso}_{\mathcal{T}}(s)$;
- $\lambda'(q, \sigma) = \lambda(q, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(q, \sigma)) \div \mathrm{mso}_{\mathcal{T}}(q)$ за $q \in Q$ и $\sigma \in \Sigma$.

Тъй като $\mathrm{mso}_{\mathcal{T}}(q) \neq 0$ за кое да е състояние $q \in Q$, делението в горната дефиниция е винаги позволено. Също така изходите на каноничната форма $\mathcal{T}'$ са от $\mathcal{R}_{[0,1]}$, защото за всяко състояние q и символ σ, такъв че $!\,\delta(q, \sigma)$,

$$\{\sigma\alpha \mid \alpha \in \Sigma^\star \wedge \delta^\star(\delta(q, \sigma), \alpha) \in F\} \subseteq \{\alpha \mid \alpha \in \Sigma^\star \wedge \delta^\star(q, \alpha) \in F\}$$

влече, че

$$\lambda(q, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(q, \sigma)) \leq \mathrm{mso}_{\mathcal{T}}(q).$$

Твърдение 5.2. *Нека* $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ *е преобразувател и* $\mathcal{T}' = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda', \iota')$ *е неговата канонична форма. Тогава*

$$(\forall q \in Q)(\forall \alpha \in \Sigma^\star)(\lambda'^\star(q, \alpha) = \lambda^\star(q, \alpha) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^\star(q, \alpha)) \div \mathrm{mso}_{\mathcal{T}}(q)).$$

Доказателство. Твърдението ще докажем с индукция по $|\alpha|$.

$|\alpha| = 0$
Тогава $\alpha = \varepsilon$ и

$$\lambda'^\star(q, \alpha) = 1 = \mathrm{mso}_{\mathcal{T}}(q) \div \mathrm{mso}_{\mathcal{T}}(q) = \lambda^\star(q, \alpha) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^\star(q, \alpha)) \div \mathrm{mso}_{\mathcal{T}}(q).$$

$|\alpha| \rightsquigarrow |\alpha| + 1$

Нека $\alpha = \alpha' a$ и твърдението е в сила за всички думи с дължина $|\alpha'|$. Тогава

$$\begin{aligned}
\lambda'^{\star}(q, \alpha) &= \lambda'^{\star}(q, \alpha' a) \\
&= \lambda'^{\star}(q, \alpha') \cdot \lambda'(\delta^{\star}(q, \alpha'), a) \\
&= (\lambda^{\star}(q, \alpha') \cdot \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(q, \alpha')) \div \mathrm{mso}_{\mathcal{T}}(q)) \cdot \\
&\quad (\lambda(\delta^{\star}(q, \alpha'), a) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(\delta^{\star}(q, \alpha'), a)) \div \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(q, \alpha'))) \\
&= \lambda^{\star}(q, \alpha') \cdot \lambda(\delta^{\star}(q, \alpha'), a) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(\delta^{\star}(q, \alpha'), a)) \div \mathrm{mso}_{\mathcal{T}}(q) \\
&= \lambda^{\star}(q, \alpha' a) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(q, \alpha' a)) \div \mathrm{mso}_{\mathcal{T}}(q) \\
&= \lambda^{\star}(q, \alpha) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(q, \alpha)) \div \mathrm{mso}_{\mathcal{T}}(q).
\end{aligned}$$

□

Твърдение 5.3. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател и $\mathcal{T}' = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda', \iota')$ е неговата канонична форма. Тогава $\mathcal{T}'$ е каноничен и $O_{\mathcal{T}} = O_{\mathcal{T}'}$.*

Доказателство. Първо, ще покажем, че преобразувателят $\mathcal{T}'$ е каноничен. За целта нека $q \in Q$ е такова, че $(\exists \alpha \in \Sigma^{\star})(\delta^{\star}(q, \alpha) \in F \wedge O^{q}_{\mathcal{T}}(\alpha) \neq 0)$. Тогава

$$\begin{aligned}
\mathrm{mso}_{\mathcal{T}'}(q) &= \max\{O^{q}_{\mathcal{T}'}(\alpha) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \\
&= \max\{\lambda'^{\star}(q, \alpha) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \\
&= \max\{\lambda^{\star}(q, \alpha) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(q, \alpha)) \div \mathrm{mso}_{\mathcal{T}}(q) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \\
&= \max\{\lambda^{\star}(q, \alpha) \div \mathrm{mso}_{\mathcal{T}}(q) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \\
&= \max\{\lambda^{\star}(q, \alpha) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \div \mathrm{mso}_{\mathcal{T}}(q) \\
&= \max\{O^{q}_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^{\star} \wedge \delta^{\star}(q, \alpha) \in F\} \div \mathrm{mso}_{\mathcal{T}}(q) \\
&= \mathrm{mso}_{\mathcal{T}}(q) \div \mathrm{mso}_{\mathcal{T}}(q) \\
&= 1.
\end{aligned}$$

Сега ще покажем, че $O_{\mathcal{T}} = O_{\mathcal{T}'}$. Ясно е, че $\mathrm{Dom}(O_{\mathcal{T}}) = \mathrm{Dom}(O_{\mathcal{T}'})$, тъй като $\mathcal{T}$ и $\mathcal{T}'$ имат една и съща функция на преходите. Нека $\alpha \in \mathrm{Dom}(O_{\mathcal{T}})$. Тогава

$$\begin{aligned}
O_{\mathcal{T}'}(\alpha) &= \iota' \cdot \lambda'^{\star}(s, \alpha) \\
&= \iota \cdot \mathrm{mso}_{\mathcal{T}}(s) \cdot \lambda^{\star}(s, \alpha) \cdot \mathrm{mso}_{\mathcal{T}}(\delta^{\star}(s, \alpha)) \div \mathrm{mso}_{\mathcal{T}}(s) \\
&= \iota \cdot \lambda^{\star}(s, \alpha) \\
&= O_{\mathcal{T}}(\alpha).
\end{aligned}$$

□

Когато използваме израза „да канонизираме преобразувател с изходи от $\mathcal{R}_{[0,1]}$", ще имаме предвид да намерим неговата канонична форма. Твърдение 5.3 ни казва, че за да канонизираме даден преобразувател $\mathcal{T}$ с изходи от $\mathcal{R}_{[0,1]}$, е достатъчно да намерим функцията $\mathrm{mso}_{\mathcal{T}}$ и да преразпределим теглата по гореописания начин.

Сега ще покажем как можем да намерим максималните изходи $\mathrm{mso}_{\mathcal{T}}(q)$ за преобразувател $\mathcal{T}$ с изходи от $\mathcal{R}_{[0,1]}$. Тази задача ще сведем до задачата за намиране на максималните изходи $\mathrm{mso}_{\mathcal{T}'}(q)$ на преобразувател $\mathcal{T}'$ с изходи от моноида $\mathcal{L}^{\infty}_{\geq 0} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, \oplus, 0)$, където $\oplus$ е операцията „събиране на реални числа" с уточнението, че $a \oplus \infty = \infty \oplus a = \infty$ за всяко $a \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. За целта първо ще покажем, че съществува моноиден изоморфизъм между $\mathcal{R}_{[0,1]}$ и $\mathcal{L}^{\infty}_{\geq 0}$.

Определение 5.5. Функцията $\rho_{\log} \colon [0, 1] \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ дефинираме за $a \in [0, 1]$ така

$$\rho_{\log}(a) = \begin{cases} \infty, & \text{ако } a = 0 \\ -\log(a), & \text{иначе} \end{cases}$$

Твърдение 5.4. *Функцията $\rho_{\log}$ е моноиден изоморфизъм между $\mathcal{R}_{[0,1]}$ и $\mathcal{L}^{\infty}_{\geq 0}$.*

Доказателство. Първо да се убедим, че $\rho_{\log}$ е хомоморфизъм.

- $\rho_{\log}(1) = -\log(1) = 0$;
- За $a, b \in [0,1]$ е в сила, че

$$\begin{aligned}
\rho_{\log}(a \cdot b) &= \begin{cases} \infty, & \text{ако } a \cdot b = 0 \\ -\log(a \cdot b), & \text{иначе} \end{cases} \\
&= \begin{cases} \infty, & \text{ако } a = 0 \vee b = 0 \\ -\log(a) + -\log(b), & \text{иначе} \end{cases} \\
&= \begin{cases} \rho_{\log}(a) \oplus \rho_{\log}(b), & \text{ако } a = 0 \vee b = 0 \\ \rho_{\log}(a) \oplus \rho_{\log}(b), & \text{иначе} \end{cases} \\
&= \rho_{\log}(a) \oplus \rho_{\log}(b).
\end{aligned}$$

Сега да видим, че $\rho_{\log}$ е инекция. Нека $a, b \in [0,1]$ и $a \neq b$. Тогава, ако $a = 0$ или $b = 0$, то $\rho_{\log}(a) \neq \rho_{\log}(b)$, тъй тако само една от двете стойности ще е равна на ∞. В противен случай, ако $a \neq 0$ и $b \neq 0$, то

$$\rho_{\log}(a) = -\log(a) \neq -\log(b) = \rho_{\log}(b).$$

Накрая ще покажем, че $\rho_{\log}$ е сюрекция. Нека $a \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. Ако $a = \infty$, то $\rho_{\log}(0) = a$, в противен случай $\exp(-a) \in [0,1]$ и

$$\rho_{\log}(\exp(-a)) = -\log(\exp(-a)) = a. \qquad \square$$

Тъй като $\rho_{\log}$ е моноиден изоморфизъм, то нейната обратна фунцкия $\rho^{-1}_{\log}$ също е моноиден изоморфизъм.

Твърдение 5.5. *Функцията $\rho^{-1}_{\log}$ е моноиден изоморфизъм между $\mathcal{L}^{\infty}_{\geq 0}$ и $\mathcal{R}_{[0,1]}$.* $\square$

Сега, използвайки $\rho_{\log}$, можем да изобразим изходите на даден преобразувател от $\mathcal{R}_{[0,1]}$ в $\mathcal{L}^{\infty}_{\geq 0}$, да намерим максималните изходи в новополучения преобразувател и от тях, посредством обратната функция на $\rho_{\log}$, да получим максималните изходи в изходния преобразувател.

Определение 5.6. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател, $\mathcal{M}'$ е моноид и $\rho\colon M \to M'$ е моноиден хомоморфизъм. *Хомоморфен преобразувател на $\mathcal{T}$ относно ρ* наричаме преобразувателя $\mathcal{T}' = (\Sigma, \mathcal{M}', Q, s, F, \delta, \lambda', \iota')$, където

- $\iota' = \rho(\iota)$;
- $\lambda'(q, a) = \rho(\lambda(q, a))$ за всяко $q \in Q$ и $a \in \Sigma$.

Твърдение 5.6. *Нека $\mathcal{T} = (\Sigma, (M, \circ_M, e_M), Q, s, F, \delta, \lambda, \iota)$ е преобразувател и $\mathcal{T}' = (\Sigma, (N, \circ_N, e_N), Q, s, F, \delta, \lambda', \iota')$ е хомоморфният преобразувател на $\mathcal{T}$ относно $\rho\colon M \to N$. Тогава е в сила, че*

$$(\forall q \in Q)(O^q_{\mathcal{T}'} = O^q_{\mathcal{T}'} \circ \rho).$$

Доказателство. Нека $q \in Q$. Ясно е, че $\mathrm{Dom}(O^q_{\mathcal{T}'}) = \mathrm{Dom}(O^q_{\mathcal{T}})$, тъй като $\mathcal{T}$ и $\mathcal{T}'$ имат едни и същи състояния и функции на преходите. Също така ρ е тотална, което означава, че $\mathrm{Dom}(O^q_{\mathcal{T}'}) = \mathrm{Dom}(O^q_{\mathcal{T}} \circ \rho)$.

Останалото ще покажем, като докажем с индукция по $|\alpha|$, че

$$(\forall \alpha \in \Sigma^\star)(\lambda'^\star(q, \alpha) = \rho(\lambda^\star(q, \alpha))).$$

$|\alpha| = 0$

Тогава $\alpha = \varepsilon$ и

$$\lambda'^\star(q, \alpha) = \lambda'^\star(q, \varepsilon) = e_N = \rho(e_M) = \rho(\lambda^\star(q, \varepsilon)) = \rho(\lambda^\star(q, \alpha)).$$

$|\alpha| \rightsquigarrow |\alpha| + 1$

Нека $\alpha = \alpha' a$ и твърдението е в сила за всяка дума с дължина $|\alpha'|$. Тогава

$$\begin{aligned}
\lambda'^\star(q, \alpha) &= \lambda'^\star(q, \alpha' a) \\
&= \lambda'^\star(q, \alpha') \circ_N \lambda'(\delta^\star(q, \alpha'), a) \\
&= \rho(\lambda^\star(q, \alpha')) \circ_N \rho(\lambda(\delta^\star(q, \alpha'), a)) \\
&= \rho(\lambda^\star(q, \alpha') \circ_M \lambda(\delta^\star(q, \alpha'), a)) \\
&= \rho(\lambda^\star(q, \alpha' a)) \\
&= \rho(\lambda^\star(q, \alpha)).
\end{aligned}$$

Сега нека $\alpha \in \mathrm{Dom}(O^q_{\mathcal{T}})$. Тогава

$$O^q_{\mathcal{T}'}(\alpha) = \lambda'^\star(q, \alpha) = \rho(\lambda^\star(q, \alpha)) = \rho(O^q_{\mathcal{T}}(\alpha)). \qquad \square$$

Сега ще дефинираме какво е максимален изход на състояние от преобразувател с изходи от $\mathcal{L}^\infty_{\geq 0}$. За целта ще въведем функцията $\widehat{\min}\colon (\mathbb{R}_{\geq 0} \cup \{\infty\}) \times (\mathbb{R}_{\geq 0} \cup \{\infty\}) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, която е дефинирана за $a, b \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ така

$$\widehat{\min}(a, b) = \begin{cases} a, & \text{ако } b = \infty \\ b, & \text{ако } a = \infty \\ \min(a, b), & \text{иначе} \end{cases}$$

където min е операцията „минимум на реални числа".

Определение 5.7. Нека $\mathcal{T} = (\Sigma, \mathcal{L}^\infty_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Дефинираме функцията $\mathrm{mso}_{\mathcal{T}}\colon Q \to \mathbb{R}_{\geq 0}$ за $q \in Q$ така

$$\mathrm{mso}_{\mathcal{T}}(q) = \begin{cases} \widehat{\min}\{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q, \alpha) \in F\}, & \text{ако } (\exists \alpha \in \Sigma^\star)(\delta^\star(q, \alpha) \in F \wedge O^q_{\mathcal{T}}(\alpha) \neq \infty) \\ 0, & \text{иначе} \end{cases}$$

От следващото твърдение и връзката, която показахме, между преобразувателите с изходи от $\mathcal{R}_{[0,1]}$ и тези с изходи от $\mathcal{L}^\infty_{\geq 0}$ следва, че „минимумът" в горната дефиниция съществува.

Твърдение 5.7. *Функцията $\rho_{\log}$ е моноиден изоморфизъм между $(\mathbb{R}_{[0,1]}, \max, 0)$ и $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \widehat{\min}, \infty)$.*

Доказателство. Вече знаем, че $\rho_{\log}$ е биекция. Остава да видим, че $\rho_{\log}$ е хомоморфизъм.

- $\rho_{\log}(0) = \infty$;
- За $a, b \in [0,1]$ е в сила, че

$$\begin{aligned}
\rho_{\log}(\max(a,b)) &= \begin{cases} \infty, & \text{ако } \max(a,b) = 0 \\ -\log(\max(a,b)), & \text{иначе} \end{cases} \\
&= \begin{cases} \infty, & \text{ако } a = 0 \wedge b = 0 \\ \min(-\log(a), -\log(b)), & \text{иначе} \end{cases} \\
&= \begin{cases} \widehat{\min}(\rho_{\log}(a), \rho_{\log}(b)), & \text{ако } a = 0 \wedge b = 0 \\ \widehat{\min}(\rho_{\log}(a), \rho_{\log}(b)), & \text{иначе} \end{cases} \\
&= \widehat{\min}(\rho_{\log}(a), \rho_{\log}(b)).
\end{aligned}$$

□

Твърдение 5.8. *Функцията $\rho_{\log}^{-1}$ е моноиден изоморфизъм между $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \widehat{\min}, \infty)$ и $(\mathbb{R}_{[0,1]}, \max, 0)$.*

Твърдение 5.9. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Нека $\mathcal{T}' = (\Sigma, \mathcal{L}^{\infty}_{\geq 0}, Q, s, F, \delta, \lambda', \iota')$ е хомоморфният преобразувател на $\mathcal{T}$ относно $\rho_{\log}$. Тогава*

$$(\forall q \in Q)(\rho_{\log}^{-1}(\mathrm{mso}_{\mathcal{T}'}(q)) = \mathrm{mso}_{\mathcal{T}}(q)).$$

Доказателство. Ако $q \in Q$ е такова, че $\neg(\exists \alpha \in \Sigma^\star)(\delta^\star(q,\alpha) \in F \wedge O^q_{\mathcal{T}'}(\alpha) \neq \infty)$, то

$$\begin{aligned}
&\neg(\exists \alpha \in \Sigma^\star)(\delta^\star(q,\alpha) \in F \wedge O^q_{\mathcal{T}}(\alpha) \neq 0), \\
&\rho_{\log}^{-1}(\mathrm{mso}_{\mathcal{T}'}(q)) = \rho_{\log}^{-1}(0) = 1 = \mathrm{mso}_{\mathcal{T}}(q).
\end{aligned}$$

Ако $q \in Q$ е такова, че $(\exists \alpha \in \Sigma^\star)(\delta^\star(q,\alpha) \in F \wedge O^q_{\mathcal{T}'}(\alpha) \neq \infty)$, то

$$\begin{aligned}
&(\exists \alpha \in \Sigma^\star)(\delta^\star(q,\alpha) \in F \wedge O^q_{\mathcal{T}}(\alpha) \neq 0), \\
\rho_{\log}^{-1}(\mathrm{mso}_{\mathcal{T}'}(q)) &= \rho_{\log}^{-1}(\widehat{\min}\{O^q_{\mathcal{T}'}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q,\alpha) \in F\}) \\
&= \max\{\rho_{\log}^{-1}(O^q_{\mathcal{T}'}(\alpha)) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q,\alpha) \in F\} \\
&= \max\{\rho_{\log}^{-1}(\rho_{\log}(O^q_{\mathcal{T}}(\alpha)) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q,\alpha) \in F\} \\
&= \max\{O^q_{\mathcal{T}}(\alpha) \mid \alpha \in \Sigma^\star \wedge \delta^\star(q,\alpha) \in F\} \\
&= \mathrm{mso}_{\mathcal{T}}(q).
\end{aligned}$$

□

До края на главата ще говорим за връзката между канонизацията на преобразувател с изходи от $\mathcal{R}_{[0,1]}$ и намирането на най-лек път в граф. Затова ще въведем няколко означения свързани с пътища в графи.

Нека $G = (V, E)$ е граф. С $\Pi_G(u, v)$ ще означаваме множеството от всички пътища в G от върха u до върха v. За $u \in V$ и $V' \subseteq V$ с $\Pi_G(u, V')$ ще означаваме множеството $\bigcup_{v \in V'} \Pi_G(u, v)$ от всички пътища от u до връх от V'. За $V', V'' \subseteq V$ с $\Pi_G(V', V'')$ ще означаваме множеството $\bigcup_{u \in V'} \Pi_G(u, V'')$. Множеството $\Pi_G(V, V)$ от всички пътища в графа G ще означаваме с Π_G.

Претеглен граф ще наричаме тройка $G = (\mathcal{M}, V, E)$, където $\mathcal{M}$ е моноид, V е крайно множество от върхове и E е крайно подмножество на $V \times M \times V$. Път в претегления граф G ще наричаме всяка редица $(v_1, \omega_1, v_2, \ldots, v_n, \omega_n, v_{n+1})$, такава че $(v_i, \omega_i, v_{i+1}) \in E$ за $1 \leq i \leq n$. Функцията $\Omega\colon \Pi_G \to \mathbb{R}$ дефинираме за $\pi \in \Pi_G$, $\pi = (v_1, \omega_1, v_2, \ldots, v_n, \omega_n, v_{n+1})$ така

$$\Omega(\pi) = \sum_{i=1}^{n} \omega_i,$$

където сумата е относно операцията на моноида $\mathcal{M}$. Стойността $\Omega(\pi)$ наричаме тегло на пътя π.

Определение 5.8. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. *δ-граф на $\mathcal{T}$* наричаме претегления граф $G_\delta = (\mathcal{M}, Q, E_\delta)$, където

$$E_\delta = \{(q, \lambda(q,\sigma), \delta(q,\sigma)) \mid (q,\sigma) \in \mathrm{Dom}(\delta)\}.$$

Очевидно е, че съществува взаимно еднозначно съответствие между пътищата в δ-графа G_δ на преобразувателя $\mathcal{T}$ с начало q и думите $\alpha \in \Sigma^\star$, с които може да бъде траверсиран $\mathcal{T}$, започвайки в състояние q, като то е такова, че теглото на път π в G_δ с начало q е равно на $\lambda^\star(q,\alpha)$, където α е думата, съответстваща на π. Тогава, когато $\mathcal{T}$ е преобразувател с изходи от $\mathcal{L}^\infty_{\geq 0}$, е очевидно и съответствието между $\mathrm{mso}_\mathcal{T}$ и теглата на най-леките пътища в G_δ.

Твърдение 5.10. *Нека $\mathcal{T} = (\Sigma, \mathcal{L}^\infty_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател и G_δ е δ-графът на $\mathcal{T}$. Тогава за всяко $q \in Q$ е в сила, че*

$$\mathrm{mso}_\mathcal{T}(q) = \begin{cases} \widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G_\delta}(q,F)\}, & \text{ако } (\exists \pi \in \Pi_{G_\delta}(q,F))(\Omega(\pi) \neq \infty) \\ 0, & \text{иначе} \end{cases}$$ □

Тоест задачата за канонизиране на преобразувател $\mathcal{T}'$ с изходи от $\mathcal{R}_{[0,1]}$ може да бъде сведена до задачата за намиране на теглата на най-леките пътища от $\Pi_{G_\delta}(q,F)$ за $q \in Q$, където $\mathcal{T} = (\Sigma, \mathcal{L}^\infty_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е хомоморфният преобразувател на $\mathcal{T}'$ относно $\rho_{\log}$, а G_δ е δ-графът на $\mathcal{T}$. Стандартно тази задачата можем да сведем до задачата за намиране на теглата на най-леките пътища от един фиксиран връх до всички останали. Нека $G'_\delta = (\mathcal{L}^\infty_{\geq 0}, Q \cup \{r\}, E'_\delta)$, където

- $r \notin Q$ е нов връх;
- $E'_\delta = \{(p,\omega,q) \mid (q,\omega,p) \in E_\delta\} \cup \{(r,0,f) \mid f \in F\}$.

Сега теглото на най-лекия път от $\Pi_{G_\delta}(q,F)$ за състояние $q \in Q$ е равно на теглото на най-лекия път от $\Pi_{G'_\delta}(r,q)$. Задачата върху графа G'_δ може да се реши ефективно с алгоритъма на Dijkstra за време $O(|E_\delta| + |Q|\log|Q|)$ (Cormen и др. 2009).

Считайки, че можем да смятаме функциите $\rho_{\log}$ и $\rho^{-1}_{\log}$ за константно време, е ясно, че графът G'_δ може да бъде построен за линейно време относно големината на изходния преобразувател $\mathcal{T}'$. Също така, след като сме намерили $\mathrm{mso}_\mathcal{T}$, можем за линейно време относно броя на δ-преходите на $\mathcal{T}'$ да променим неговите изходи. Така получаваме, че времето за канонизация на преобразувател с изходи от $\mathcal{R}_{[0,1]}$ се мажорира от времето за намиране на теглата на най-леките пътища от върха r до останалите върхове в графа G'_δ.

Твърдение 5.11. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{[0,1]}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Тогава можем да построим каноничната форма на $\mathcal{T}$ за време*

$$O(|\,\mathrm{Dom}(\delta)| + |Q|\log|Q|).$$ □

Решаването на задачата за намиране на максималните изходи на състоянията в преобразувател с изходи от $\mathcal{L}^\infty_{\geq 0}$ вместо от $\mathcal{R}_{[0,1]}$ е по-удобно не само защото можем да използваме алгоритъма на Dijkstra, но и защото от практическа гледна точка е „по-евтино" да събираме, а не да умножаваме „реални числа". Другата практическа изгода е, че се избягва недостатъка на числата с плаваща запетая, които губят бързо точност около нулата.

5.2 Канонизация на f-преобразуватели с $\mathrm{Rng}(\lambda_f) \subseteq [0,1]$

Сега ще разгледаме задачата за канонизация на f-преобразуватели, чиито подлежащи преобразуватели са с изходи от $\mathcal{R}_{[0,1]}$, тоест λ_f е функция от $Q \times \Sigma$ в $[0,1]$. До края на този раздел под f-преобразувател ще разбираме f-преобразувател, който притежава това свойство.

Определение 5.9. f-преобразувател наричаме *каноничен*, ако подлежащият му преобразувател е каноничен.

Определение 5.10. Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател и $\mathcal{T}$ е подлежащият му преобразувател. *Канонична форма* на $\mathcal{F}$ наричаме f-преобразувателя $\mathcal{F}' = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda', \varphi', \iota')$, където

- $\iota' = \iota \cdot \mathrm{mso}_{\mathcal{T}}(s)$;
- $\lambda'(q,\sigma) = \lambda(q,\sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(q,\sigma)) \div \mathrm{mso}_{\mathcal{T}}(q)$ за $(q,\sigma) \in \mathrm{Dom}(\delta)$;
- $\varphi'(q) = \varphi(q) \cdot \mathrm{mso}_{\mathcal{T}}(f(q)) \div \mathrm{mso}_{\mathcal{T}}(q)$ за $q \in \mathrm{Dom}(\varphi)$.

Тъй като неявно изискваме $\mathcal{T}$ да е с изходи от $\mathcal{R}_{[0,1]}$ и знаем, че $\mathrm{mso}_{\mathcal{T}}(q) \neq 0$ за кое да е състояние на $\mathcal{T}$, деленията в горната дефиниция са позволени.

Твърдение 5.12. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател и $\mathcal{T}$ е подлежащият му преобразувател. Нека $\mathcal{F}' = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda', \varphi', \iota')$ е каноничната форма на $\mathcal{F}$. Тогава*

$$(\forall q \in Q)(\forall \sigma \in \Sigma)(\lambda'_f(q,\sigma) = \lambda_f(q,\sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q,\sigma)) \div \mathrm{mso}_{\mathcal{T}}(q)).$$

Доказателство. От $\mathrm{Dom}(\lambda') = \mathrm{Dom}(\lambda)$ следва, че $\mathrm{Dom}(\lambda'_f) = \mathrm{Dom}(\lambda_f) = \mathrm{Dom}(\delta_f)$, тоест е достатъчно да покажем верността на твърдението само за двойките $(q,\sigma) \in \mathrm{Dom}(\delta_f)$. Нека $(q,\sigma) \in \mathrm{Dom}(\delta_f)$. Тогава е ясно, че съществува редица от състояния $q_0, q_1, \ldots, q_{m+1}$, такава че $q_0 = q$, $q_{i+1} = f(q_i)$ и $\neg!\,\delta(q_i,\sigma)$ за $0 \leq i \leq m-1$ и $q_{m+1} = \delta(q_m,\sigma)$. [1]

С индукция по i ще покажем, че

$$(\forall 0 \leq i \leq m)(\lambda'_f(q_{m-i},\sigma) = \lambda_f(q_{m-i},\sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q_{m-i},\sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_{m-i})),$$

което при $i = m$ представлява точно твърдението, което доказваме.

$i = 0$

Тогава $q_{m-i} = q_m$ и тъй като $\delta(q_m,\sigma) = q_{m+1}$,

$$\begin{aligned}\lambda'_f(q_m,\sigma) &= \lambda'(q_m,\sigma)\\ &= \lambda(q_m,\sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta(q_m,\sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_m)\\ &= \lambda_f(q_m,\sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q_m,\sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_m).\end{aligned}$$

$i \rightsquigarrow i+1$

Нека твърдението е в сила за $0 \leq i \leq m-1$. Тогава за $i+1$ е в сила, че $\neg!\,\delta(q_{m-(i+1)},\sigma)$, $f(q_{m-(i+1)}) = q_{m-i}$ и

$$\lambda'_f(q_{m-(i+1)},\sigma) = \varphi'(q_{m-(i+1)}) \cdot \lambda'_f(q_{m-i},\sigma)$$

1. В противен случай би било вярно, че $\neg!\,\delta(f^{(k)}(q),\sigma)$ и $!\,\delta_f(f^{(k)}(q),\sigma)$ за всяко $k \in \mathbb{N}$. Но това означава, че съществува цикъл $f^{(0)}(q), f^{(1)}(q), \ldots, f^{(m+1)}(q)$, такъв че $\neg!\,\delta(f^{(i)}(q),\sigma)$ и $!\,\delta_f(q_i,\sigma)$ за $0 \leq i \leq m+1$, което противоречи с минималността относно включване на δ_f – минималната функция не би била дефинирана за никое състояние от този цикъл и буквата σ.

$$
\begin{aligned}
&= (\varphi(q_{m-(i+1)}) \cdot \mathrm{mso}_{\mathcal{T}}(q_{m-i}) \div \mathrm{mso}_{\mathcal{T}}(q_{m-(i+1)}))\cdot \\
&\quad (\lambda_f(q_{m-i}, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q_{m-i}, \sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_{m-i})) \\
&= \varphi(q_{m-(i+1)}) \cdot \lambda_f(q_{m-i}, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q_{m-i}, \sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_{m-(i+1)}) \\
&= \lambda_f(q_{m-(i+1)}, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q_{m-(i+1)}, \sigma)) \div \mathrm{mso}_{\mathcal{T}}(q_{m-(i+1)}). \qquad \square
\end{aligned}
$$

Твърдение 5.13. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател и $\mathcal{F}' = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda', \varphi', \iota')$ е неговата канонична форма. Тогава $\mathcal{F}'$ е каноничен и $O_{\mathcal{F}} = O_{\mathcal{F}'}$.*

Доказателство. Нека $\mathcal{T}$ и $\mathcal{T}'$ са подлежащите преобразуватели съответно на $\mathcal{F}$ и $\mathcal{F}'$. Тогава $\mathcal{T}'$ има начален изход ι' и функция на изходите λ'_f. Съгласно определение 5.10 и твърдение 5.12

- $\iota' = \iota \cdot \mathrm{mso}_{\mathcal{T}}(s)$;
- $\lambda'_f(q, \sigma) = \lambda_f(q, \sigma) \cdot \mathrm{mso}_{\mathcal{T}}(\delta_f(q, \sigma)) \div \mathrm{mso}_{\mathcal{T}}(q)$ за $q \in Q$ и $\sigma \in \Sigma$.

Това означава, че $\mathcal{T}'$ е каноничната форма на $\mathcal{T}$, тъй като $\mathcal{T}$ има начален изход ι и функция на изходите λ_f. Тоест $\mathcal{T}'$ е каноничен преобразувател, от което следва, че $\mathcal{F}'$ е каноничен f-преобразувател. От твърдение 5.3 и твърдение 3.2 получаваме, че

$$O_{\mathcal{F}'} = O_{\mathcal{T}'} = O_{\mathcal{T}} = O_{\mathcal{F}}. \qquad \square$$

Когато използваме израза „да канонизираме f-преобразувател", ще имаме предвид да намерим неговата канонична форма. От предходното твърдение става ясно, че можем да канонизираме даден f-преобразувател за време асимптотично равно на времето за намиране на максималните изходи на състоянията на неговия подлежащ преобразувател.

Твърдение 5.14. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Тогава можем да построим каноничната форма на $\mathcal{F}$ за време*

$$O(|\operatorname{Dom}(\delta_f)| + |Q| \log |Q|). \qquad \square$$

5.3 Канонизация на f-преобразуватели на ниво думи

От твърдение 3.5 следва, че f-преобразувателите на ниво думи за изгладени n-грамни езикови модели имат подлежащи преобразуватели с изходи от $\mathcal{R}_{[0,1]}$. Следователно от предходния раздел знаем, че f-преобразувателите на ниво думи за изгладени n-грамни езикови модели могат да бъдат канонизирани за време $O(|Q||\Sigma| + |Q| \log |Q|)$.

Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки, $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпус над $\widetilde{\Sigma}_1$ и $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\}) \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Ще въведем следните означения с цел облекчаване на записа.

- $|\mathcal{C}| = \sum_{i=1}^{m} |\alpha_i|$;
- $|\rho(\widetilde{\Sigma}_1 \cup \{\$\})| = \sum_{a \in \widetilde{\Sigma}_1 \cup \{\$\}} |\rho(a)|$;
- $|\rho(\mathcal{C})| = \sum_{i=1}^{m} |\rho(\alpha_i)|$.

Следващото твърдение получаваме като директно следствие от твърдение 3.7 и твърдение 5.14.

Твърдение 5.15. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F}$ е f-преобразувателят на ниво думи за $P^{(n)}$. Тогава можем да построим каноничната форма на $\mathcal{F}$ за време*

$$O(n|\mathcal{C}|(|\widetilde{\Sigma}| + \log(n|\mathcal{C}|))).$$ □

Причината, поради която f-преобразувателите на ниво думи могат да имат изходи по-големи от 1, е, че константите e_α от дефиницията на изгладен n-грамен езиков модел могат да бъдат произволни неотрицателни реални числа. Разглеждането само на изгладени n-грамни езикови модели, чиито константи e_α принадлежат на интервала $[0, 1]$, не е много рестриктивно. Всички примери за изгладени n-грамни езикови модели от част 1.5 отговарят на това условие. Всъщност всички изглаждащи техники, които използват интерполация могат да бъдат представени с константи e_α, ненадминаващи 1.

Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$, $n \in \mathbb{N}^+$ и $\widehat{P^{(i)}}$ за $1 \leq i \leq n$ е i-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\widehat{P^{(i)}}(a \mid \alpha) = 0$ за $a \in \widetilde{\Sigma}$ и $\alpha \in \{h_i(\beta) \mid \beta \in \widetilde{\Sigma}^\star \circ \{\$\}\}$, такива че $\#_{\mathcal{C}}(\alpha a) = 0$. Тогава всички n-грамни езикови модели над $\{\hat{}\} \circ \widetilde{\Sigma} \circ \{\$\}$, чиито параметри са дефинирани за $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \widetilde{\Sigma}^\star \circ \{\$\}\}$ така

$$P^{(n)}(a \mid \alpha) = \lambda_\beta \widehat{P^{(|\beta|+1)}}(a \mid \beta) + (1 - \lambda_\beta) P^{(|\beta|)}(a \mid \beta'),$$

където $\beta = g_{\mathcal{C}}(\alpha)$, $\beta' \succeq \beta$, $|\beta'| = |\beta| - 1$, $\lambda_\beta \in [0, 1]$ и $P^{(|\beta|)}$ е $|\beta|$-грамен езиков модел от същия тип, са изгладени n-грамни езикови модели с константи e_α, принадлежащи на интервала $[0, 1]$. Наистина, избирайки $d_{\alpha a} = \lambda_\alpha \widehat{P^{(n)}}(a \mid \alpha) + (1 - \lambda_\alpha) P^{(n-1)}(a \mid \alpha')$ и $e_\alpha = 1 - \lambda_\alpha$ за $a \in \widetilde{\Sigma} \cup \{\$\}$, $\alpha \in \{h_n(\beta) \mid \beta \in \widetilde{\Sigma}^\star \circ \{\$\}\}$, $\alpha' \succeq \alpha$, $|\alpha'| = |\alpha| - 1$, получаваме, че за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$ и $\alpha \in \{h_n(\beta) \mid \beta \in \widetilde{\Sigma}^\star \circ \{\$\}\}$

$$P^{(n)}(a \mid \alpha) = \begin{cases} P^{(|\beta|+1)}(a \mid \beta), & \text{ако } |\beta| < n - 1 \\ d_{\alpha a}, & \text{ако } |\beta| = n - 1 \wedge \#_{\mathcal{C}}(\alpha a) > 0 \\ e_\alpha P^{(n-1)}(a \mid \alpha'), & \text{иначе} \end{cases}$$

където $\beta = g_{\mathcal{C}}(\alpha)$, $\alpha' \succeq \alpha$ и $|\alpha'| = |\alpha| - 1$.

Тъй като този вид изгладени n-грамни езикови модели обхваща голяма част от езиковите модели, които се използват на практика (Chen и Goodman 1998), ще покажем как могат да се канонизират по-ефективно техните представяния чрез f-преобразуватели на ниво думи.

Канонизацията на f-преобразуватели на ниво думи за изгладени n-грамни езикови модели, чиито изходи са от $\mathcal{R}_{[0,1]}$, може да се направи по-сръчно, като максималните изходи на състоянията на подлежащия преобразувател се намерят чрез прилагане на алгоритъма на Dijkstra върху граф, който е по-малък от δ-графа на подлежащия преобразувател. Този граф ще дефинираме за произволен f-преобразувател на ниво думи или символи, тъй като впоследствие ще го използваме и за други цели. Така теглата на ребрата на този граф ще бъдат елементи на моноида $\mathcal{L}^\infty = (\mathbb{R} \cup \{\infty\}, \oplus, \infty)$, където $\oplus$ е операцията „събиране на реални числа" с добавката, че $a \oplus \infty = \infty \oplus a = \infty$ за всяко $a \in \mathbb{R} \cup \{\infty\}$.

Определение 5.11. Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво думи или символи за изгладен n-грамен езиков модел. С $G(\mathcal{F})$ ще означаваме претегления граф $(\mathcal{L}^\infty, V, E)$, където

- $V = Q \cup \{(q, f(q)) \mid q \in \mathrm{Dom}(f)\}$;
- $E = E_0 \cup E_1 \cup E_2 \cup E_3$;

- $E_0 = \{(q, \rho_{\log}(\lambda(q,\sigma)), \delta(q,\sigma)) \mid (q,\sigma) \in \mathrm{Dom}(\delta)\}$;
- $E_1 = \{(q, \rho_{\log}(\varphi(q)), (q, f(q))) \mid q \in \mathrm{Dom}(f)\}$;
- $E_2 = \{((q,f(q)), \rho_{\log}(\lambda(f(q),\sigma)), \delta(f(q),\sigma)) \mid q \in \mathrm{Dom}(f) \wedge (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}$;
- $E_3 = \{((q,f(q)), \rho_{\log}(\varphi(f(q))), (f(q), f(f(q)))) \mid q \in \mathrm{Dom}(f) \wedge f(q) \in \mathrm{Dom}(f)\}$.

Първо ще покажем, че графът $G(\mathcal{F})$ за f-преобразувател $\mathcal{F}$ на ниво думи или символи за изгладен n-грамен езиков модел има не повече ребра от δ-графа на подлежащия преобразувател на $\mathcal{F}$. Всъщност това, което се вижда от доказателството е, че в този граф се „разгръщат" не повече състояния, отколкото в подлежащия преобразувател. Под „разгръщане" на състояние имаме предвид допълването на δ-преходите на състоянието с преходите, описани чрез δ_f.

Твърдение 5.16. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво думи или символи за изгладен n-грамен езиков модел $P^{(n)}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава*

- $|V| \in O(|Q|)$;
- $|E| \in O(|Q| + |\mathrm{Dom}(\delta)| + |L||\Sigma|)$,

където $L = \{q \in \mathrm{Dom}(f) \mid \neg(\exists p \in Q)(f(p) = q)\}$.

Доказателство. Твърдението за броя на върховете $|V|$ е ясно, тъй като

$$|V| = |Q| + |\{(q, f(q)) \mid q \in \mathrm{Dom}(f)\}| = |Q| + |\mathrm{Dom}(f)| \leq 2|Q| \in O(|Q|).$$

За броя на преходите е ясно, че

$$\begin{aligned} |E_0| &= |\mathrm{Dom}(\delta)|, \\ |E_1| &= |\mathrm{Dom}(f)| \leq |Q|, \\ |E_3| &\leq |\mathrm{Dom}(f)| \leq |Q|. \end{aligned}$$

Ще покажем, че $|E_2| \leq |L||\Sigma|$. Тъй като f-графът $G_f = (Q, E_f)$ на $\mathcal{F}$ е ацикличен, G_f е гора и върховете от L са листата на дърветата в G_f, които не са корени. [2] Нека $G'_f = ((\mathbb{N}, +, 0), Q, E'_f)$ е гората G_f, на чиито ребра сме дали тегла по следния начин

$$\begin{aligned} E'_f &= \{(q, \eta(q,p), p) \mid (q,p) \in E_f\}, \\ \eta(q,p) &= |\{\sigma \mid (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (p,\sigma) \in \mathrm{Dom}(\delta)\}| \text{ за } (q,p) \in E_f. \end{aligned}$$

Ясно е, че $(q, \eta(q,p), p) \in E'_f \iff (q,p) \in E_f \iff (q,p) = (q, f(q))$. Тогава

$$\begin{aligned} |E_2| &= |\{((q,f(q)), \rho_{\log}(\lambda(f(q),\sigma)), \delta(f(q),\sigma)) \mid q \in \mathrm{Dom}(f) \wedge (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \\ &\leq |\{((q,f(q)),\sigma) \mid q \in \mathrm{Dom}(f) \wedge (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \\ &= |\{((q,f(q)),\sigma) \mid (q, \eta(q,f(q)), f(q)) \in E'_f \wedge (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \\ &= |\bigcup_{(q,\eta(q,f(q)),f(q)) \in E'_f} \{((q,f(q)),\sigma) \mid (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \end{aligned}$$

2. В случая на ребрата в графа гледаме като на стрелки от наследниците към предшествениците, тоест листа са тези върхове, които нямат входящо ребро.

$$
\begin{aligned}
&= \sum_{(q,\eta(q,f(q)),f(q))\in E'_f} |\{((q,f(q)),\sigma) \mid (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \\
&= \sum_{(q,\eta(q,f(q)),f(q))\in E'_f} |\{\sigma \mid (q,\sigma) \notin \mathrm{Dom}(\delta) \wedge (f(q),\sigma) \in \mathrm{Dom}(\delta)\}| \\
&= \sum_{(q,\eta(q,f(q)),f(q))\in E'_f} \eta(q,f(q)) \\
&\leq \sum_{\pi\in\Pi_{G'_f}(L,Q\setminus\mathrm{Dom}(f))} \Omega(\pi).
\end{aligned}
$$

Корените на дърветата от гората G'_f са тези състояния, които нямат излизащи f-преходи. Следователно последното равенство е вярно, тъй като всяко ребро от G'_f участва в поне един път от листо, което не е корен, (тоест от връх от L) до корен на дърво от G'_f (тоест до връх от $Q \setminus \mathrm{Dom}(f)$).

Тъй като G'_f е ацикличен, $|\Pi_{G'_f}(l, Q \setminus \mathrm{Dom}(F))| = 1$ за всяко $l \in L$. Нека

$$\pi_l = (q_1^l, \eta(q_1^l, q_2^l), q_2^l, \ldots, q_{n_l}^l, \eta(q_{n_l}^l, q_{n_l+1}^l), q_{n_l+1}^l)$$

е единственият път от $\Pi_{G'_f}(l, Q \setminus \mathrm{Dom}(F))$ за $l \in L$. Тогава, използвайки свойствата на f-преобразувателите на ниво думи и ниво символи, формулирани в твърдение 3.9 и твърдение 4.14, получаваме

$$
\begin{aligned}
|E_2| &\leq \sum_{l\in L} \Omega(\pi_l) \\
&= \sum_{l\in L} \sum_{i=1}^{n_l} \eta(q_i^l, q_{i+1}^l) \\
&= \sum_{l\in L} \sum_{i=1}^{n_l} |\{\sigma \mid (q_i^l,\sigma) \notin \mathrm{Dom}(\delta) \wedge (q_{i+1}^l,\sigma) \in \mathrm{Dom}(\delta)\}| \\
&= \sum_{l\in L} \sum_{i=1}^{n_l} |\{\sigma \mid (q_{i+1}^l,\sigma) \in \mathrm{Dom}(\delta)\}| - |\{\sigma \in \Sigma \mid (q_i^l,\sigma) \in \mathrm{Dom}(\delta)\}| \\
&= \sum_{l\in L} |\{\sigma \mid (q_{n_l+1}^l,\sigma) \in \mathrm{Dom}(\delta)\}| - |\{\sigma \in \Sigma \mid (q_1^l,\sigma) \in \mathrm{Dom}(\delta)\}| \\
&= \sum_{l\in L} |\{\sigma \mid (q_1^l,\sigma) \notin \mathrm{Dom}(\delta) \wedge (q_{n_l+1}^l,\sigma) \in \mathrm{Dom}(\delta)\}| \\
&\leq \sum_{l\in L} |\Sigma| \\
&= |L||\Sigma|.
\end{aligned}
$$

□

Твърдение 5.17. *Нека $\mathcal{C} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за $P^{(n)}$. Тогава*

$$|\{q \in \mathrm{Dom}(f) \mid \neg(\exists p \in Q)(f(p) = q)\}| \leq |\mathcal{C}|.$$

Доказателство. По дефиниция

$$Q = \{q \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |q| < n \wedge \#_{\mathcal{C}}(q) > 0\}.$$

Следователно за всяко $q \in Q$ съществуват $1 \leq k \leq m$ и $0 \leq i \leq j \leq |\alpha_k|$, такива че $q = a^k_{i+1}a^k_{i+2}\dots a^k_j$, където с a^k_l за $1 \leq l \leq |\alpha_k|$ означаваме l-тия символ на α_k.

Да допуснем, че

$$|\{q \in \mathrm{Dom}(f) \mid \neg(\exists p \in Q)(f(p) = q)\}| > |\mathcal{C}|.$$

Тогава съществуват две различни състояния $q_1, q_2 \in \mathrm{Dom}(f) \subseteq Q$, такива че

$$\neg(\exists p \in Q)(f(p) = q_1),$$
$$\neg(\exists p \in Q)(f(p) = q_2),$$

за които съществуват $1 \leq k \leq m$ и $0 \leq i_1 < i_2 \leq j \leq |\alpha_k|$, такива че, без ограничения на общността,

$$q_1 = a^k_{i_1+1}a^k_{i_1+2}\dots a^k_j,$$
$$q_2 = a^k_{i_2+1}a^k_{i_2+2}\dots a^k_j.$$

Нека α е състояние в $\mathcal{F}$ и $\neg!\, f(\alpha)$. Тогава е вярно, че $(\forall \beta \in \mathrm{Suf}(\{\alpha\}))(\neg!\, f(\beta))$. Наистина, щом $\neg!\, f(\alpha)$, то $(\forall a \in \widetilde{\Sigma} \cup \{\$\})(!\, \delta(\alpha, a))$. Тогава, съгласно твърдение 3.9,

$$(\forall \beta \in \mathrm{Suf}(\{\alpha\}))(\forall a \in \widetilde{\Sigma} \cup \{\$\})(!\, \delta(\beta, a)),$$

което означава, че $(\forall \beta \in \mathrm{Suf}(\{\alpha\}))(\neg!\, f(\beta))$.

Сега щом $!\, f(q_2)$, то $!\, f(a^k_m a^k_{m+1}\dots a^k_j)$ за всяко $i_1 + 1 \leq m \leq i_2 + 1$. Тогава $f(q_{i_2}) = q_{i_2+1}$, което е абсурд. Следователно

$$|\{q \in \mathrm{Dom}(f) \mid \neg(\exists p \in Q)(f(p) = q)\}| \leq |\mathcal{C}|. \qquad \square$$

Твърдение 5.18. *Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки и $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Нека $\mathcal{C} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ е корпус над $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава*

$$|\{q \in Q \mid \neg(\exists p \in Q)(f(p) = q)\}| \leq |\rho(\mathcal{C})|.$$

Доказателство. По дефиниция

$$Q = \{(q', a') \mid q' \in Q' \wedge a' \in \mathrm{Pref}(\{\rho(a) \mid (q', a) \in \mathrm{Dom}(\delta')\}) \wedge \# \not\geq a'\},$$
$$Q' = \{q' \in (\widetilde{\Sigma} \cup \{\hat{}, \$\})^\star \mid |q'| < n \wedge \#_{\mathcal{C}}(q') > 0\},$$

където $(\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$.

Следователно за всяко $(\alpha, \beta) \in Q$ съществуват $1 \leq k \leq m$ и $0 \leq i \leq j \leq |\alpha_k|$, такива че $\alpha = a^k_{i+1}a^k_{i+2}\dots a^k_j$ и $\beta \preceq \rho(a^k_j)$, където с a^k_l за $1 \leq l \leq |\alpha_k|$ означаваме l-тия символ на α_k. Това от своя страна означава, че за всяко $(\alpha, \beta) \in Q$ съществуват $1 \leq k \leq m$ и $0 \leq i \leq j \leq |\rho(\alpha_k)|$, такива че $\rho(\alpha)\beta = b^k_{i+1}b^k_{i+2}\dots b^k_j$, където с b^k_l за $1 \leq l \leq |\rho(\alpha_k)|$ означаваме l-тия символ на $\rho(\alpha_k)$.

Да допуснем, че

$$|\{q \in Q \mid \neg(\exists p \in Q)(f(p) = q)\}| > |\rho(\mathcal{C})|.$$

Тогава съществуват две различни състояния $(\alpha_1, \beta_1), (\alpha_2, \beta_2) \in \mathrm{Dom}(f) \subseteq Q$, такива че

$$\neg(\exists p \in Q)(f(p) = (\alpha_1, \beta_1)),$$

$$\neg(\exists p \in Q)(f(p) = (\alpha_2, \beta_2)),$$

за които съществуват $1 \leq k \leq m$ и $0 \leq i_1 \leq i_2 \leq j \leq |\rho(\alpha_k)|$, такива че, без ограничения на общността,

$$\rho(\alpha_1)\beta_1 = b^k_{i_1+1} b^k_{i_1+2} \dots b^k_j,$$
$$\rho(\alpha_2)\beta_2 = b^k_{i_2+1} b^k_{i_2+2} \dots b^k_j.$$

Това означава, че $\rho(\alpha_2)\beta_2 \succeq \rho(\alpha_1)\beta_1$, от което лесно се съобразява, тъй като ρ е разделител, че $\beta_1 = \beta_2$ и $\alpha_2 \succeq \alpha_1$. Но тогава щом $!\, f((\alpha_2, \beta_2))$, то $!\, f'(\alpha_2)$ и повтаряйки разсъжденията от предходното твърдение, заключаваме, че $(\exists \gamma \in Q')(f'(\gamma) = \alpha_2)$. Това от своя страна влече, че за това γ е в сила, че $f((\gamma, \beta_2)) = (\alpha_2, \beta_2)$, което е абсурд. Следователно

$$|\{q \in Q \mid \neg(\exists p \in Q)(f(p) = q)\}| \leq |\rho(\mathcal{C})|. \qquad \square$$

Сега ще се убедим, че за всеки път в f-преобразувател $\mathcal{F}$ на ниво думи или символи за изгладен n-грамен езиков модел от състояние q до състояние p съществува път в $\Pi_{G(\mathcal{F})}(q, p)$ с тегло равно на $\rho_{\log}$ от теглото на пътя във f-преобразувателя. Тук под $\rho_{\log}$ имаме предвид естественото разширение на функцията $\rho_{\log} \colon [0,1] \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ до $\rho_{\log} \colon \mathbb{R}_{\geq 0} \to \mathbb{R} \cup \{\infty\}$.

Твърдение 5.19. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво думи или символи за изгладения n-грамен езиков модел $P^{(n)}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава за всяко $(q, a) \in \mathrm{Dom}(\delta_f)$ съществува път $\pi \in \Pi_{G(\mathcal{F})}(q, \delta_f(q, a))$, $\pi = (v_1, \omega_1, v_2, \dots, v_{m+1}, \omega_{m+1}, v_{m+2})$, такъв че*

$$(\forall 1 \leq i \leq m)((v_i, \omega_i, v_{i+1}) \notin E_0 \cup E_2) \wedge (v_{m+1}, \omega_{m+1}, v_{m+2}) \in E_0 \cup E_2 \wedge \Omega(\pi) = \rho_{\log}(\lambda_f(q, a)).$$

Доказателство. Твърдението ще докажем с индукция по $\mathrm{level}_f(q)$.

$\mathbf{level_f(q) = 0}$

Тогава $(q, a) \in \mathrm{Dom}(\delta)$, $(q, \rho_{\log}(\lambda(q, a)), \delta(q, a)) \in E_0$ и

$$\Omega((q, \rho_{\log}(\lambda(q, a)), \delta(q, a))) = \rho_{\log}(\lambda(q, a)) = \rho_{\log}(\lambda_f(q, a)).$$

$\mathbf{level_f(q) \rightsquigarrow level_f(q) + 1}$

Нека $(q, a) \in \mathrm{Dom}(\delta_f)$, $\mathrm{level}_f(q) > 0$ и твърдението е в сила за всяко $(p, a) \in \mathrm{Dom}(\delta_f)$, такова че $\mathrm{level}_f(p) = \mathrm{level}_f(q) - 1$. Ако $(q, a) \in \mathrm{Dom}(\delta)$, можем да повторим разсъжденията от базовия случай. Затова нека $(q, a) \notin \mathrm{Dom}(\delta)$. Тогава $q \in \mathrm{Dom}(f)$, $!\, \delta_f(f(q), a)$ и $(q, \rho_{\log}(\varphi(q)), (q, f(q))) \in E_1$. От индукционното предположение знаем, че съществува път $\pi' \in \Pi_{G(\mathcal{F})}(f(q), \delta_f(f(q), a))$, $\pi' = (v_1, \omega_1, v_2, \dots, v_{m+1}, \omega_{m+1}, v_{m+2})$, такъв че

$$(\forall 1 \leq i \leq m)((v_i, \omega_i, v_{i+1}) \notin E_0 \cup E_2),$$
$$(v_{m+1}, \omega_{m+1}, v_{m+2}) \in E_0 \cup E_2,$$
$$\Omega(\pi') = \rho_{\log}(\lambda_f(f(q), a)).$$

Тъй като $v_1 \in Q$, то $(v_1, \omega_1, v_2) \in E_0 \cup E_1$.

Ако $(v_1, \omega_1, v_2) \in E_0$, то $(v_1, \omega_1, v_2) = (f(q), \rho_{\log}(\lambda(f(q), \sigma)), \delta(f(q), \sigma))$ за $\sigma \in \Sigma$. Тогава това е последното ребро на π', тоест $m = 0$ и $\sigma = a$, защото в противен случай $v_2 \neq \delta(f(q), a)$. [3] Щом $(f(q), \rho_{\log}(\lambda(f(q), a)), \delta(f(q), a)) \in E_0$, то

$$((q, f(q)), \rho_{\log}(\lambda(f(q), a)), \delta(f(q), a)) \in E_2,$$

3. Щом q има f-преход, то $n \geq 2$. За f-преобразувателите на ниво думи и ниво символи за изгладени n-грамни езикови модели при $n \geq 2$ е в сила, че ако $(q, a), (q, b) \in \mathrm{Dom}(\delta)$ и $a \neq b$, то $\delta(q, a) \neq \delta(q, b)$. Това е така, защото $a \succeq \delta(q, a)$ и $b \succeq \delta(q, b)$.

$$\Omega((v_1, \omega_1, v_2)) = \Omega(((q, f(q)), \rho_{\log}(\lambda(f(q), a)), \delta(f(q), a))).$$

Следователно за пътя $\pi = (v'_0, \omega'_0, v'_1, \omega'_1, v'_2)$, където $v'_0 = q$, $\omega'_0 = \rho_{\log}(\varphi(q))$, $v'_1 = (q, f(q))$, $\omega'_1 = \omega_1$ и $v'_2 = v_2$, е вярно, че само последното му ребро е от $E_0 \cup E_2$ и

$$\begin{aligned}\Omega(\pi) &= \Omega((v'_0, \omega'_0, v'_1)) \oplus \Omega((v'_1, \omega'_1, v'_2)) \\ &= \rho_{\log}(\varphi(q)) \oplus \rho_{\log}(\lambda(f(q), a)) \\ &= \rho_{\log}(\varphi(q) \cdot \lambda(f(q), a)) \\ &= \rho_{\log}(\lambda_f(q, a)).\end{aligned}$$

Ако $(v_1, \omega_1, v_2) \in E_1$, тоест $v_1 = f(q)$, $\omega_1 = \rho_{\log}(\varphi(f(q)))$ и $v_2 = (f(q), f(f(q)))$, то

$$((q, f(q)), \rho_{\log}(\varphi(f(q))), (f(q), f(f(q)))) \in E_3$$
$$\Omega((v_1, \omega_1, v_2)) = \Omega(((q, f(q)), \rho_{\log}(\varphi(f(q))), (f(q), f(f(q))))).$$

Тогава за пътя $\pi = (v'_0, \omega'_0, v'_1, \ldots, v'_{m+1}, \omega'_{m+1}, v'_{m+2})$, където $v'_0 = q$, $\omega'_0 = \rho_{\log}(\varphi(q))$, $v'_1 = (q, f(q))$ и $v'_i = v_i$ и $\omega'_{i-1} = \omega_{i-1}$ за $2 \leq i \leq m+2$, е вярно, че само последното му ребро е от $E_0 \cup E_2$ и

$$\begin{aligned}\Omega(\pi) &= \Omega((v'_0, \omega'_0, v'_1)) \oplus \Omega(\pi') \\ &= \rho_{\log}(\varphi(q)) \oplus \rho_{\log}(\lambda_f(f(q), a)) \\ &= \rho_{\log}(\varphi(q) \cdot \lambda_f(f(q), a)) \\ &= \rho_{\log}(\lambda_f(q, a)).\end{aligned}$$

□

Сега ще се убедим, че за всеки път в $\Pi_{G(\mathcal{F})}(q, p)$ съществува път в f-преобразувателя $\mathcal{F}$ на ниво думи или символи за изгладен n-грамен езиков модел от състояние q до състояние p с тегло равно на $\rho_{\log}^{-1}$ от теглото на пътя в графа.

Твърдение 5.20. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи или символи за изгладения n-грамен езиков модел $P^{(n)}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава за всяко $q \in Q$ и $\pi \in \Pi_{G(\mathcal{F})}(q, Q)$, $\pi = (v_1, \omega_1, v_2, \ldots, v_{m+1}, \omega_{m+1}, v_{m+2})$, такъв че*

$$(\forall 1 \leq i \leq m)((v_i, \omega_i, v_{i+1}) \notin E_0 \cup E_2) \wedge (v_{m+1}, \omega_{m+1}, v_{m+2}) \in E_0 \cup E_2,$$

съществува $a \in \Sigma$, такова че

$$\delta_f(q, a) = v_{m+2} \wedge \rho_{\log}(\lambda_f(q, a)) = \Omega(\pi).$$

Доказателство. Твърдението ще докажем с индукция по m.

$m = 0$

Тогава $\pi = (v_1, \omega_1, v_2)$ и $(v_1, \omega_1, v_2) \in E_0$. Тоест $(v_1, \omega_1, v_2) = (q, \rho_{\log}(\lambda(q, a)), \delta(q, a))$ за някое $a \in \Sigma$. Тогава за това a е в сила, че $\delta_f(q, a) = \delta(q, a) = v_2$ и

$$\rho_{\log}(\lambda_f(q, a)) = \rho_{\log}(\lambda(q, a)) = \Omega((v_1, \omega_1, v_2)) = \Omega(\pi).$$

$m \rightsquigarrow m + 1$

Нека $m > 0$ и твърдението е в сила за всеки път с дължина $m - 1$, удовлетворяващ предпоставките на твърдението. Нека сега $\pi = (v_1, \omega_1, v_2)\pi'$ удовлетворява предпоставките и $\pi' \in \Pi_{G(\mathcal{F})}(v_2, v_{m+2})$, $\pi' = (v_2, \omega_2, v_3, \ldots, v_{m+1}, \omega_{m+1}, v_{m+2})$. Щом $m > 0$, то

$(v_1, \omega_1, v_2) \notin E_0 \cup E_2$. Знаем, че $v_1 \in Q$, което влече, че $(v_1, \omega_1, v_2) \in E_1$, тоест $v_2 \in V \setminus Q$. Всеки връх от $V \setminus Q$ има излизащи ребра само от $E_2 \cup E_3$. Тъй като само последното ребро $(v_{m+1}, \omega_{m+1}, v_{m+2})$ е от $E_0 \cup E_2$, то всички ребра на π освен първото и последното са от E_3. Следователно

- $v_{i+1} = (f^{(i-1)}(v_1), f^{(i)}(v_1))$ за $1 \leq i \leq m$;
- $\omega_i = \rho_{\log}(\varphi(f^{(i-1)}(v_1)))$ за $1 \leq i \leq m$;
- $(v_{m+1}, \omega_{m+1}, v_{m+2}) = ((f^{(m-1)}(v_1), f^{(m)}(v_1)), \rho_{\log}(\lambda(f^{(m)}(v_1), a)), \delta(f^{(m)}(v_1), a))$ за някое $a \in \Sigma$.

Знаем, че за всяко ребро, излизащо от $(v_1, f(v_1))$, съществува ребро със същото тегло и край, излизащо от $f(v_1)$. Следователно пътят $\pi'' = (v_2'', \omega_2'', v_3'', \ldots, v_{m+1}'', \omega_{m+1}'', v_{m+2}'')$, където $v_2'' = f(v_1)$ и $v_i'' = v_i$ и $\omega_{i-1}'' = \omega_{i-1}$ за $3 \leq i \leq m+2$, е такъв, че само последното му ребро е от $E_0 \cup E_2$. От индукционното предположение следва, че съществува $a \in \Sigma$ (същото от дефиницията на $(v_{m+1}, \omega_{m+1}, v_{m+2})$), [4] такова че

$$\delta_f(f(v_1), a) = v_{m+2}'' = v_{m+2} \wedge \rho_{\log}(\lambda_f(f(v_1), a)) = \Omega(\pi'') = \Omega(\pi').$$

Да допуснем, че $!\,\delta(v_1, a)$. Тогава, съгласно твърдение 3.9 и твърдение 4.14, получаваме, че $!\,\delta(f^{(k)}(v_1), a)$ за $0 \leq k \leq m$. Това противоречи със съществуването на реброто

$$(v_{m+1}, \omega_{m+1}, v_{m+2}) = ((f^{(m-1)}(v_1), f^{(m)}(v_1)), \rho_{\log}(\lambda(f^{(m)}(v_1), a)), \delta(f^{(m)}(v_1), a)).$$

Следователно $\neg\,!\,\delta(v_1, a)$, $\delta_f(v_1, a) = \delta_f(f(v_1), a) = v_{m+2}$ и

$$\begin{aligned}
\rho_{\log}(\lambda_f(v_1, a)) &= \rho_{\log}(\varphi(v_1) \cdot \lambda_f(f(v_1), a)) \\
&= \rho_{\log}(\varphi(v_1)) \oplus \rho_{\log}(\lambda_f(f(v_1), a)) \\
&= \Omega((v_1, \omega_1, v_2)) \oplus \Omega(\pi') \\
&= \Omega(\pi).
\end{aligned}$$

□

От твърдение 5.19 и твърдение 5.20 следват лесно с индукция, съответно по дължината на думата и дължината на пътя, следващите две твърдения.

Твърдение 5.21. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи или символи за изгладения n-грамен езиков модел $P^{(n)}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава за всяко $(q, \alpha) \in \mathrm{Dom}(\delta_f^\star)$ съществува път $\pi \in \Pi_{G(\mathcal{F})}(q, \delta_f^\star(q, \alpha))$, такъв че*

$$\Omega(\pi) = \rho_{\log}(\lambda_f^\star(q, \alpha)).$$

□

Твърдение 5.22. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи или символи за изгладения n-грамен езиков модел $P^{(n)}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава за всяко $q \in Q$ и всеки път $\pi \in \Pi_{G(\mathcal{F})}(q, Q)$, $\pi = (v_1, v_2, \ldots, v_{m+2})$ съществува $\alpha \in \Sigma^\star$, такова че*

$$\delta_f^\star(q, \alpha) = v_{m+2} \wedge \rho_{\log}(\lambda_f^\star(q, \alpha)) = \Omega(\pi).$$

□

Директно от предходните две твърдения следва, че максималният изход на състояние $q \in Q$ на подлежащия преобразувател на f-преобразувателя $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ на ниво думи за изгладен n-грамен езиков модел е равен на образа на теглото на най-лекия път от $\Pi_{G(\mathcal{F})}(q, F)$ под действието на $\rho_{\log}^{-1}$. Тъй като всяко състояние q в f-преобразувател $\mathcal{F}$ на ниво думи за изгладен n-грамен езиков модел има път с ненулево тегло до финално състояние, в графа $G(\mathcal{F})$ винаги ще има път от $\Pi_{G(\mathcal{F})}(q, F)$ с тегло различно от ∞.

4. Щом разглеждаме f-преходи, означава, че $n > 1$. В този случай това a ще е еднозначно определено и ще бъде последната буква от състоянието v_{m+2}.

Твърдение 5.23. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\mathcal{T}$ е подлежащият преобразувател на $\mathcal{F}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^{\infty}, V, E)$. Тогава за всяко $q \in Q$ е в сила, че*

$$\mathrm{mso}_{\mathcal{T}}(q) = \rho_{\log}^{-1}(\widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})}(q, F)\}).$$

Ако $\mathcal{F}$ е f-преобразувател на ниво думи за изгладен n-грамен езиков модел с изходи от $\mathcal{R}_{[0,1]}$, графът $G(\mathcal{F})$ ще бъде с тегла от $\mathcal{L}^{\infty}_{\geq 0}$, което означава, че можем да прилагаме алгоритъма на Dijkstra върху него. Тогава от предходното твърдение следва, че можем да намерим максималните изходи на състоянията на подлежащия преобразувател на $\mathcal{F}$, от графа $G(\mathcal{F})$.

За размера на $G(\mathcal{F})$ за f-преобразувател $\mathcal{F}$ на ниво думи можем да дадем оценка, използвайки твърдение 3.7, твърдение 5.16 и твърдение 5.17.

Твърдение 5.24. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Тогава графът $G(\mathcal{F}) = (\mathcal{L}^{\infty}, V, E)$ за f-преобразувателя на ниво думи $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е такъв, че*

- $|V| \in O(n|\mathcal{C}|)$;
- $|E| \in O((|\widetilde{\Sigma}| + n)|\mathcal{C}|)$. □

Като следствие можем да дадем оценка за времето за канонизация на f-преобразувател на ниво думи с изходи от $\mathcal{R}_{[0,1]}$.

Твърдение 5.25. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F} = (\widetilde{\Sigma} \cup \{\$\}, \mathcal{R}_{[0,1]}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво думи за $P^{(n)}$. Тогава можем да построим каноничната форма на $\mathcal{F}$ за време*

$$O(|\widetilde{\Sigma}||\mathcal{C}| + n|\mathcal{C}| \log(n|\mathcal{C}|)).$$ □

5.4 Канонизация на f-преобразуватели на ниво символи

f-преобразувателите на ниво символи за изгладени n-грамни езикови модели могат да имат изходи по f-преходите по-големи от 1 поради същата причина, поради която f-преобразувателите на ниво думи за изгладени n-грамни езикови модели могат да имат f-преходи с изходи извън $[0, 1]$. При f-преобразувателите на ниво символи, обаче, не е сигурно, че изходите на подлежащите им преобразуватели са от моноида $\mathcal{R}_{[0,1]}$.

Затова ще разширим понятията „максимален изход на състояние от преобразувател", „каноничен преобразувател" и „каноничен f-преобразувател" за случаите, в които изходите на преобразувателите и f-преобразувателите са от $\mathcal{R}_{\geq 0}$. Максимален изход $\mathrm{mso}_{\mathcal{T}}(q)$ на състояние q от преобразувател $\mathcal{T}$ с изходи от $\mathcal{R}_{\geq 0}$ дефинираме по същия начин. [5] Преобразувател $\mathcal{T}$ с изходи от $\mathcal{R}_{\geq 0}$ наричаме каноничен, ако максималните изходи на всички негови състояния съществуват и са равни на 1. f-преобразувател с изходи от $\mathcal{R}_{\geq 0}$ наричаме каноничен, ако подлежащият му преобразувател е каноничен.

Имайки дефиницията за максимален изход на състояние на преобразувател с изходи от $\mathcal{R}_{\geq 0}$ можем да отбележим, че е в сила аналогично твърдение на твърдение 5.23. То е директно следствие от твърдение 5.21 и твърдение 5.22.

5. В този по-общ случай е възможно максимумът от дефиницията да не съществува

Твърдение 5.26. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво символи за изгладения n-грамен езиков модел $P^{(n)}$. Нека $\mathcal{T}$ е подлежащият преобразувател на $\mathcal{F}$ и $G(\mathcal{F})$ е претегленият граф $(\mathcal{L}^\infty, V, E)$. Тогава за всяко $q \in Q$ е в сила, че*

$$\mathrm{mso}_{\mathcal{T}}(q) = \rho_{\log}^{-1}(\widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})}(q, F)\}).$$

Можем да забележим, че конструкцията на каноничната форма на даден f-преобразувател, описана в предходния раздел, не зависи съществено от това, че подлежащият преобразувател е с изходи от $\mathcal{R}_{[0,1]}$. Всъщност за конструкцията е достатъчно максималните изходите на състоянията на подлежащия преобразувател да са дефинирани. Тоест, ако покажем, че максималните изходи на състоянията на подлежащия преобразувател на f-преобразувател на ниво символи съществуват, то можем да приложим същата конструкция, за да намерим каноничен f-преобразувател, еквивалентен на изходния.

Състоянията на един f-преобразуател $\mathcal{F}$ на ниво символи, построен от f-преобразувател $\mathcal{F}'$ на ниво думи, можем да разделим на две групи – такива с втора координата равна на ε и съответстващи на състояния от $\mathcal{F}'$ и такива, които съответстват на разбиването на преходите на $\mathcal{F}'$ на символи. Ще разгледаме поотделно как можем да намерим максималните изходи на тези две групи състояния.

Твърдение 5.27. *Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувател на ниво думи за изгладен n-грамен езиков модел $P^{(n)}$ и $\mathcal{T}'$ е неговият подлежащ преобразувател. Нека $\mathcal{F} = (\Sigma_1 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво символи за $P^{(n)}$ с разделител ρ и $\mathcal{T}$ е неговият подлежащ преобразувател. Тогава*

$$(\forall \alpha \in Q')(\mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon)) = \mathrm{mso}_{\mathcal{T}'}(\alpha)).$$

Доказателство. Нека $\alpha \in Q'$. Можем да повторим разсъжденията от твърдение 4.10 за състоянието (α, ε) и да заключим, че

- $\mathrm{Dom}(O_{\mathcal{F}}^{(\alpha,\varepsilon)}) = \{\rho(\alpha) \mid \alpha \in \widetilde{\Sigma}_1^\star \circ \{\$\}\}$;
- $(\forall \beta \in \widetilde{\Sigma}_1^\star)(O_{\mathcal{F}}^{(\alpha,\varepsilon)}(\rho(\beta\$)) = O_{\mathcal{F}'}^{\alpha}(\beta\$))$.

Това означава, че

$$\{O_{\mathcal{F}}^{(\alpha,\varepsilon)}(\beta) \mid \beta \in \mathrm{Dom}(O_{\mathcal{F}}^{(\alpha,\varepsilon)})\} = \{O_{\mathcal{F}'}^{\alpha}(\beta) \mid \beta \in \mathrm{Dom}(O_{\mathcal{F}'}^{\alpha})\},$$

тоест $\mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon)) = \mathrm{mso}_{\mathcal{T}'}(\alpha)$. □

Тъй като вече показахме, че можем да намираме максималните изходи на състоянията на подлежащия преобразувател на произволен f-преобразувател на ниво думи за изгладен n-грамен езиков модел, получаваме метод за намиране на максималните изходи на тези състояния от f-преобразувател на ниво символи, които имат втора координата равна на ε.

Максималните изходи на останалите състояния ще намерим, като се възползваме от това, че знаем максималните изходи на състоянията с втора координата ε.

Твърдение 5.28. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател на ниво символи за изгладен n-грамен езиков модел и $\mathcal{T}$ е неговият подлежащ преобразувател. Нека $G(\mathcal{F})$ е графът $(\mathcal{L}^\infty, V, E)$ и $G(\mathcal{F})'$ е графът $(\mathcal{L}^\infty, V', E')$, където*

- $V' = V \cup \{r\}$ *и* $r \notin V$;
- $E' = E \setminus \{e \in E \mid \mathrm{Proj}_2(\mathrm{Proj}_1(e)) = \varepsilon\} \cup \{((\alpha, \varepsilon), \mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon)), r) \mid (\alpha, \varepsilon) \in Q\}$.

Тогава е в сила, че

- $(\forall q \in Q)(\widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})'}(q, r)\} = \widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})}(q, F)\})$;
- $G(\mathcal{F})'$ *е ацикличен.*

Доказателство. Първо ще отбележим, че от всяко състояние на f-преобразувател на ниво символи започва път, който завършва във финално състояние. Това означава, че в графа $G(\mathcal{F})'$ ще има път от всяко състояние до състоянието r, а в графа $G(\mathcal{F})$ от всяко състояние до състояние от F. Също така най-леките пътища отразени чрез „минимумите" съществуват. Това е така, тъй като всяко състояние $q \in Q$ е достижимо от състоние $q' \in Q$ с втора координата равна на ε, а за тях вече знаем, че „минимумите" съществуват.

Нека $q \in Q$. Ако $\mathrm{Proj}_2(q) = \varepsilon$, твърдението е очевидно вярно. Затова нека $\mathrm{Proj}_2(q) \neq \varepsilon$. Да допуснем, че съществува път $\pi \in \Pi_{G(\mathcal{F})'}(q, r)$, такъв че $\Omega(\pi) < \widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})}(q, F)\}$. Тогава $\pi = \pi'\pi''$, където π' е път с начало q и край (α, ε) и не съдържа новия връх r, а π'' е пътят $((\alpha, \varepsilon), \mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon)), r)$. Това означава, че най-късият път от $\Pi_{G(\mathcal{F})}((\alpha, \varepsilon), F)$ има тегло $\mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon))$. Но тогава π е свидетел за път в $\Pi_{G(\mathcal{F})}(q, F)$ с тегло $\Omega(\pi)$, което противоречи с допускането. Нека сега да допуснем, че съществува път $\pi \in \Pi_{G(\mathcal{F})}(q, F)$, такъв че $\Omega(\pi) < \widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})'}(q, r)\}$. π задължително съдържа връх $q' \in Q$, такъв че $\mathrm{Proj}_2(q') = \varepsilon$ (най-малкото последният връх от пътя е такъв). Нека (α, ε) е първият такъв връх в π, тоест $\pi = \pi'\pi''$, където $\pi' \in \Pi_{G(\mathcal{F})}(q, (\alpha, \varepsilon))$ и $\pi'' \in \Pi_{G(\mathcal{F})}((\alpha, \varepsilon), F)$. Ясно е, че това са най-леките пътища от съответните множества. Тогава $\Omega(\pi'') = \mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon))$. Също така, тъй като π' се състои само от върхове с втора координата различна от ε, пътят $\pi'((\alpha, \varepsilon), \mathrm{mso}_{\mathcal{T}}((\alpha, \varepsilon)), r)$ е път от $\Pi_{G(\mathcal{F})'}(q, r)$ с тегло $\Omega(\pi)$. Това противоречи на допускането. Така получаваме, че

$$\widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})'}(q, r)\} = \widehat{\min}\{\Omega(\pi) \mid \pi \in \Pi_{G(\mathcal{F})}(q, F)\}.$$

За втората част от твърдението да допуснем, че $(v_0, \omega_0, v_1, \ldots, \omega_{m-1}, v_m, \omega_m, v_{m+1})$, $v_0 = v_{m+1}$ е цикъл в $G(\mathcal{F})'$. Ясно е, че този цикъл не съдържа връх с втора координата ε, тъй като те имат изходящи ребра само към r, който от своя страна няма нито едно изходящо ребро. Следователно всички върхове v_i за $0 \leq i \leq m$ са такива, че $\mathrm{Proj}_2(v_i) \neq \varepsilon$. Тогава, съгласно дефиницията на $G(\mathcal{F})$, върховете от този цикъл принадлежат на едно от множествата

$$V_1 = \{(\alpha, \beta) \in Q \mid \beta \neq \varepsilon\},$$
$$V_2 = \{((\alpha, \beta), (\alpha', \beta)) \in Q \times Q \mid (\alpha', \beta) = f((\alpha, \beta))\}.$$

Без ограничения на общността можем да считаме, че $v_0 \in V_1$. В случай че $v_0 \in V_2$ и $v_0 = ((\alpha, \beta), (\alpha', \beta))$, тъй като изходящите ребра на $((\alpha, \beta), (\alpha', \beta))$ са подмножество на изходящите ребра на (α', β), можем да разгледаме същия цикъл с разликата, че първият връх е заменен с (α', β).

С индукция по i ще докажем, че за всяко $1 \leq i \leq m + 1$

$$v_i \in V_1 \implies |\mathrm{Proj}_1(v_i)| \leq |\mathrm{Proj}_1(v_0)| \wedge |\mathrm{Proj}_2(v_i)| > |\mathrm{Proj}_2(v_0)|,$$
$$v_i \in V_2 \implies |\mathrm{Proj}_1(\mathrm{Proj}_2(v_i))| < |\mathrm{Proj}_1(v_0)| \wedge |\mathrm{Proj}_2(\mathrm{Proj}_2(v_i))| \geq |\mathrm{Proj}_2(v_0)|.$$

$i = 1$

Съгласно дефиницията на $G(\mathcal{F})$, $v_0 \in V_1$, $v_0 = (\alpha, \beta)$ има изходящи ребра от вида

$$((\alpha, \beta), \rho_{\log}(\lambda((\alpha, \beta), b)), (\alpha, \beta b)),$$
$$((\alpha, \beta), \rho_{\log}(\varphi((\alpha, \beta))), ((\alpha, \beta), (\alpha', \beta))),$$

където $\alpha' \succeq \alpha$, $|\alpha'| = |\alpha| - 1$ и $b \in \Sigma$. И в двата случая виждаме, че твърдението е изпълнено.

$\boldsymbol{i \rightsquigarrow i+1}$

Нека твърдението е в сила за $1 \leq i \leq m$. Нека първо да разгледаме случая, в който $v_i \in V_1$. Тогава, съгласно индукционното предположение, $v_i = (\alpha', \beta')$, $v_0 = (\alpha, \beta)$, $|\alpha'| \leq |\alpha|$ и $|\beta'| > |\beta|$. От дефиницията на $G(\mathcal{T})$ следва, че изходящите от v_i ребра имат вида

$$((\alpha', \beta'), \rho_{\log}(\lambda((\alpha', \beta'), b)), (\alpha', \beta' b)),$$
$$((\alpha', \beta'), \rho_{\log}(\varphi((\alpha', \beta'))), ((\alpha', \beta'), (\alpha'', \beta'))),$$

където $\alpha'' \succeq \alpha$, $|\alpha''| = |\alpha'| - 1$ и $b \in \Sigma$. В първия случай би било вярно, че $|\alpha'| \leq |\alpha|$ и $|\beta' b| > |\beta|$, а във втория случай, че $|\alpha''| < |\alpha|$ и $|\beta'| \geq |\beta|$.

Нека сега да разгледаме случая, в който $v_i \in V_2$. Тогава, съгласно индукционното предположение, $v_i = (\gamma, (\alpha', \beta'))$, $v_0 = (\alpha, \beta)$, $|\alpha'| < |\alpha|$ и $|\beta'| \geq |\beta|$. От дефиницията на $G(\mathcal{T})$ следва, че изходящите от v_i ребра имат вида

$$((\gamma, (\alpha', \beta')), \rho_{\log}(\lambda((\alpha', \beta'), b)), (\alpha', \beta' b)),$$
$$((\gamma, (\alpha', \beta')), \rho_{\log}(\varphi((\alpha', \beta'))), ((\alpha', \beta'), (\alpha'', \beta'))),$$

където $\alpha'' \succeq \alpha'$, $|\alpha''| = |\alpha'| - 1$ и $b \in \Sigma$. В първия случай би било вярно, че $|\alpha'| \leq |\alpha|$ и $|\beta' b| > |\beta|$, а във втория случай, че $|\alpha''| < |\alpha|$ и $|\beta'| \geq |\beta|$.

За $i = m + 1$ получаваме, че $|\operatorname{Proj}_2(v_{m+1})| = |\operatorname{Proj}_2(v_0)| > |\operatorname{Proj}_2(v_0)|$, което е абсурд. Следователно графът $G(\mathcal{F})'$ е ацикличен. □

Предходното твърдение ни казва, че за да намерим максималните изходи на състоянията от f-преобразувател $\mathcal{F}$ на ниво символи, които имат втора координата различна от ε, е достатъчно да намерим най-леките пътища в $G(\mathcal{F})'$ от всяко такова състояние до новия връх r. Графът $G(\mathcal{F})'$ има отрицателни тегла и затова не можем да приложим алгоритъма на Dijkstra. Той, обаче, е ацикличен, което означава, че можем да намерим теглата на най-леките пътища до r с динамично програмиране за линейно време относно размера на графа (Cormen и др. 2009). Размерът на $G(\mathcal{F})'$ е асимптотично равен на размера на $G(\mathcal{F})$, за чийто размер можем дадем оценка, използвайки твърдение 4.11, твърдение 5.16 и твърдение 5.18.

Твърдение 5.29. *Нека $\widetilde{\Sigma}_1$ и Σ_2 са азбуки и $\rho\colon (\widetilde{\Sigma}_1 \cup \{\$\})^\star \to (\Sigma_2 \cup \{\#\})^\star$ е разделител. Нека $\mathcal{C}$ е корпус над $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изграден n-грамен езиков модел за $\mathcal{C}$. Тогава графът $G(\mathcal{F}) = (\mathcal{L}^\infty, V, E)$ за f-преобразувателя на ниво символи $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ за $P^{(n)}$ с разделител ρ е такъв, че*

- $|V| \in O(|\rho(\widetilde{\Sigma}_1 \cup \{\$\})| + n|\rho(\mathcal{C})|)$;
- $|E| \in O(|\rho(\widetilde{\Sigma}_1 \cup \{\$\})| + n|\rho(\mathcal{C})| + |\Sigma_2||\rho(\mathcal{C})|)$. □

Като следствие можем да дадем оценка за времето за канонизация на f-преобразувател на ниво символи в случаите, в които съответстващият му f-преобразувател на ниво думи е с изходи от $\mathcal{R}_{\geq 0}$ или $\mathcal{R}_{[0,1]}$.

Твърдение 5.30. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изграден n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{\geq 0}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава можем да построим каноничната форма на $\mathcal{F}$ за време*

$$O(n|\mathcal{C}|(|\widetilde{\Sigma}_1| + \log(n|\mathcal{C}|)) + |\rho(\mathcal{C})|(|\Sigma_2| + n) + |\rho(\widetilde{\Sigma}_1 \cup \{\$\})|).$$

Доказателство. Както вече споменахме, достатъчно за конструкцията на каноничната форма е да знаем максималните изходи на състоянията на подлежащия преобразувател на $\mathcal{F}$. Максималните изходи на състоянията $q \in Q$, такива че $\mathrm{Proj}_2(q) = \varepsilon$, можем да намерим, съгласно твърдение 5.27 и твърдение 5.15, за време

$$O(n|\mathcal{C}|(|\widetilde{\Sigma}_1| + \log(n|\mathcal{C}|))).$$

Максималните изходи на състоянията $q \in Q$, такива че $\mathrm{Proj}_2(q) \neq \varepsilon$, можем да намерим, съгласно твърдение 5.28 и твърдение 5.29 за време

$$O(|\rho(\mathcal{C})|(|\Sigma_2| + n) + |\rho(\widetilde{\Sigma}_1 \cup \{\$\})|).$$

Считайки, че можем да смятаме функциите $\rho_{\log}$ и $\rho_{\log}^{-1}$ за константно време, получаваме, че f-преобразувателят $\mathcal{F}$ може да бъде канонизиран за време

$$O(n|\mathcal{C}|(|\widetilde{\Sigma}_1| + \log(n|\mathcal{C}|)) + |\rho(\mathcal{C})|(|\Sigma_2| + n) + |\rho(\widetilde{\Sigma}_1 \cup \{\$\})|). \qquad \square$$

Твърдение 5.31. *Нека $\mathcal{C}$ е корпус над азбуката $\widetilde{\Sigma}_1$ и $P^{(n)}$ е изгладен n-грамен езиков модел за $\mathcal{C}$. Нека $\mathcal{F}' = (\widetilde{\Sigma}_1 \cup \{\$\}, \mathcal{R}_{[0,1]}, Q', s', F', \delta', f', \lambda', \varphi', \iota')$ е f-преобразувателят на ниво думи за $P^{(n)}$ и $\mathcal{F} = (\Sigma_2 \cup \{\#\}, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувателят на ниво символи за $P^{(n)}$ с разделител ρ. Тогава можем да построим каноничната форма на $\mathcal{F}$ за време*

$$O(|\widetilde{\Sigma}_1||\mathcal{C}| + n|\mathcal{C}|\log(n|\mathcal{C}|) + |\rho(\mathcal{C})|(|\Sigma_2| + n) + |\rho(\widetilde{\Sigma}_1 \cup \{\$\})|).$$

Доказателство. Доказателството е същото като това от предходното твърдение с разликата, че в този случай можем, съгласно твърдение 5.27 и твърдение 5.25, да намерим максималните изходи на състоянията $q \in Q$, такива че $\mathrm{Proj}_2(q) = \varepsilon$ за време

$$O(|\widetilde{\Sigma}_1||\mathcal{C}| + n|\mathcal{C}|\log(n|\mathcal{C}|)). \qquad \square$$

5.5 Псевдокод за канонизация на f-преобразуватели, представящи езикови модели

Сега ще представим псевдокод за методите, представени в тази глава, за канонизация на f-преобразуватели на ниво думи и ниво символи за изгладени n-грамни езикови модели.

5.5.1 Псевдокод за канонизация на f-преобразуватели на ниво думи

Първо ще опишем с псевдокод канонизацията на f-преобразувател на ниво думи. В алгоритъм 5.1 е представена функцията CanonicalWordFTransducer, която построява каноничната форма на дадения f-преобразувател на ниво думи. Изходното представяне на f-преобразувателя е същото като представянето, което връща функцията WordFTransducer. Във функцията CanonicalWordFTransducer първо се намират максималните изходи на състоянията на хомоморфния образ на подлежащия преобразувател на изходния f-преобразувател относно $\rho_{\log}$. Ще покажем как това може да стане на време $O(|f||\Sigma| + |f|\log(|f|))$, ако изходният f-преобразувател има изходи по-големи от 1, и $O(|f||\Sigma| \div n + |f|\log(|f|))$ в противен случай, където n е редът на езиковия модел, представен чрез f-преобразувателя. Останалите два цикъла от функцията следват строго конструкцията за канонична форма на f-преобразувател и отнемат линейно време относно размера на изходния

Алгоритъм 5.1. Построява каноничната форма на дадения f-преобразувател на ниво думи.

```
1: function CanonicalWordFTransducer((s, δ, f, λ, φ))
2:     mso ← WordFTransducerMso((s, δ, f, λ, φ))
3:     λ' ← λ
4:     for all (q, a, r) ∈ λ' do
5:         r' ← r + mso[HashSearch(δ, (q, a))] − mso[q]
6:         HashInsert(λ', (q, a), exp(−r'))
7:     φ' ← φ
8:     for all i ← 1, |f| do
9:         r' ← φ[i] + mso[f[i]] − mso[i]
10:        φ'[i] ← exp(−r')
11:    return (s, δ, f, λ', φ')
```

f-преобразувател. Така функцията CanonicalWordFTransducer има времева сложност $O(|f||\Sigma|+|f|\log(|f|))$ или $O(|f||\Sigma| \div n+|f|\log(|f|))$ в зависимост от дадения f-преобразувател.

В алгоритъм 5.2 е представена функцията WordFTransducerMso, която намира максималните изходи на състоянията на хомоморфния образ на подлежащия преобразувател на дадения f-преобразувател на ниво думи относно $\rho_{\log}$. Това става като първо се построява граф, в който теглата на най-леките пътища (изобразени чрез $\rho_{\log}^{-1}$) отговарят на максималните изходи на състоянията. Ако даденият f-преобразувател има f-преходи с тегла по-големи от 1, това е δ-графът на хомоморфния образ на подлежащия преобразувател на изходния f-преобразувател относно $\rho_{\log}$. Този граф бива построен посредством функцията UnderlyingLogGraph за линейно време относно неговия размер (знаем, че броят на върховете му е $O(|f|)$, а броят на ребрата му е $O(|f||\Sigma|)$). В противен случай графът е $G(\mathcal{F})$, където $\mathcal{F}$ е изходният f-преобразувател. Този граф бива построен посредством функцията FTransducerGraph за линейно време относно неговия размер (знаем, че броят на върховете му е $O(|f|)$, а броят на ребрата му е $O(|f| + |f||\Sigma| \div n)$). Проверката на ред 3 може да бъде имплементирана за линейно време относно броя на f-преходите. Функцията ReverseGraph обръща посоката на ребрата на графа за линейно време относно неговата големина, а функцията Dijkstra намира теглата на най-късите пътища от зададения като първи аргумент връх до всеки един от останали върхове в графа. Времевата сложност на функцията Dijkstra е $O(|E|+|V|\log|V|)$, където E са ребрата, а V са върховете на дадения граф. Можем да забележим, че тъй като в псевдокода за конструкциите на f-преобразувател на ниво думи и ниво символи съобразихме, че ни е нужно само едно финално състояние, в случая не се нуждаем от добавяне на ново състояние, което да обединява всички финални. Тази роля играе състоянието с номер 2 – това е номерът на финалното състояние на дадения f-преобразувател. Така, ако изходният f-преобразувател няма изходи по-големи от 1, функцията WordFTransducerMso ще отнеме $O(|f||\Sigma| \div n + |f|\log(|f|))$ време. В противен случай ще отнеме $O(|f||\Sigma| + |f|\log(|f|))$ време.

Функцията UnderlyingLogGraph от алгоритъм 5.3 построява δ-графа на хомоморфния образ на подлежащия преобразувател на дадения f-преобразувател относно $\rho_{\log}$. Резултатният граф ще има по точно $|\Sigma|$ ребра, излизащи от всеки връх. Затова бива представен чрез матрицата E с размери $|f| \times |\Sigma|$. Всеки елемент $E[q][i]$ ще представлява двойка (r, p) и ще носи семантиката, че i-тото ребро на върха q има тегло r и влиза във върха p. На ред 2 се намира размера на азбуката на f-преобразувателя (предполагаме, че буквите в азбуката са последователни и започват от 1). Това може да стане за $O(|\delta|)$, като се разгледат всички преходи на f-преобразувателя. След това се построява матрицата E и се намира

Алгоритъм 5.2. Намира максималните изходи на състоянията на хомоморфния образ на подлежащия преобразувател на дадения f-преобразувател на ниво думи относно $\rho_{\log}$.

```
1: function WordFTransducerMso((s, δ, f, λ, φ))
2:     graph ← NIL
3:     if HasOutputGreaterThan1(φ) then
4:         graph ← UnderlyingLogGraph((s, δ, f, λ, φ))
5:     else
6:         graph ← FTransducerGraph((s, δ, f, λ, φ))
7:     return Dijkstra(2, ReverseGraph(graph))
```

топологическата сортировка на състоянията на f-преобразувателя относно функцията на f-преходите. Тъй като f-графът на всеки f-преобразувател на ниво думи е ацикличен, това може да се постигне за такива f-преобразуватели за линейно време относно броя на техните състояния. Тоест $|T| = |f|$ и за всяко $1 \leq i \leq |T|$ е в сила, че ако $f[T[i]] \neq 0$ (тоест $!\, f(T[i])$), то индексът на $f[T[i]]$ в T ще бъде по-малък от i. **for** цикълът от ред 5 разглежда състоянията в топологически ред и за всяко от тях добавя всичките му $|\Sigma|$ на брой ребра в графа, представен чрез E. За дадено състояние q ребрата са или такива, които съответстват на δ-преходи от q (разглеждат се на ред 10), или такива, които съответстват на δ_f-преход, който не е δ-преход (те се разглеждат на ред 12). Съществото е, че тъй като състоянията се разглеждат в топологически ред, всеки път, когато се изпълни ред 12, реброто $E[f[q]][\sigma]$ вече ще е било добавено. Също така, съгласно дефиницията на f-преобразувател на ниво думи, ако се изпълни този ред, то задължително състоянието q има f-преход. Времевата и пространствената сложност на UnderlyingLogGraph са очевидно $O(|f||\Sigma|)$.

Алгоритъм 5.3. Построява δ-графа на хомоморфния образ на подлежащия преобразувател на дадения f-преобразувател относно $\rho_{\log}$.

```
1: function UnderlyingLogGraph((s, δ, f, λ, φ))
2:     alphabet ← AlphabetSize(δ)
3:     E ← NewMatrix(|f|, alphabet, NIL)
4:     T ← TopologicalSort(f)
5:     for i ← 1, |T| do
6:         q ← T[i]
7:         for σ ← 1, alphabet do
8:             r ← HashSearch(λ, (q, σ))
9:             if r ≠ NIL then
10:                E[q][σ] ← (− log(r), p)
11:            else
12:                E[q][σ] ← (E[f[q]][σ][1] − log(φ[q]), E[f[q]][σ][2])
13:    return
```

В алгоритъм 5.4 е представена функцията FTransducerGraph, която по даден f-преобразувател $\mathcal{F}$, представящ езиков модел на ниво думи или символи, построява претегления граф $G(\mathcal{F})$. Построеният граф е представен чрез списъци на съседство, тоест неговите върхове са представени чрез естествените числа в интервала $[1, |E|]$ и $E[i]$ за $1 \leq i \leq |E|$ е масив от тройки, всяка от които описва едно ребро в $G(\mathcal{F})$. По-точно – всеки елемент на $E[i]$ за $1 \leq i \leq |E|$ е тройка (σ, ω, j), такава че (i, ω, j) е ребро в $G(\mathcal{F})$, а σ е входната буква, съответстваща на реброто, ако $(i, \omega, j) \in E_0 \cup E_2$, или 0 в противен случай.

Алгоритъм 5.4. По даден f-преобразувател $\mathcal{F}$, представящ езиков модел на ниво думи или символи, построява претегления граф $G(\mathcal{F})$.

```
1:  function FTransducerGraph((s, δ, f, λ, φ))
2:      E ← NewArray(|f|, NewArray())
3:      for all (q, a, p) ∈ δ do
4:          ArrayAppend(E[q], (a, − log HashSearch(λ, (q, a)), p))
5:      F ← NewArray(|f|, 0)
6:      T ← TopologicalSort(f)
7:      for i ← 1, |T| do
8:          q ← T[i]
9:          p ← f[q]
10:         if p ≠ 0 then
11:             F[q] ← |E| + 1
12:             ArrayAppend(E[q], (0, φ[q], F[q]))
13:             ArrayAppend(E, NewArray())
14:             for j ← 1, |E[p]| do
15:                 (input, output, target) ← E[p][j]
16:                 if HashSearch(δ, (q, input)) = NIL then
17:                     ArrayAppend(E[F[q]], (input, output, target))
18:             if f[p] ≠ 0 then
19:                 ArrayAppend(E[F[q]], (0, φ[p], F[p]))
20:     return E
```

Сега ще обясним как алгоритъм 5.4 съответства на дефиницията на $G(\mathcal{F})$. **for** цикълът от ред 3 добавя към E всички ребра на графа от E_0. Това са ребрата, които отговарят на δ-преходи в f-преобразувателя. На ред 5 дефинираме нов масив, в който на позиция $1 \leq i \leq |f|$ ще записваме номера на новия връх $(i, f(i))$, съответстващ на състоянието i. Тъй като знаем, че f-графите на f-преобразувателите на ниво думи и ниво символи са ациклични, на ред 6 намираме за $O(|f|)$ време топологическата сортировка на състоянията на f-преобразувателя относно релацията на f-преходите. Тоест $|T| = |f|$ и за всяко $1 \leq i \leq |T|$ е в сила, че ако $f[T[i]] \neq 0$ (тоест $!\, f(T[i])$), то индексът на $f[T[i]]$ в T ще бъде по-малък от i. Сега за всяко състояние q от ред 8, такова че има f-преход, тоест $p \neq 0$ от ред 10,

- на ред 11 запазваме в $F[q]$ номера на новия връх $(q, f(q))$;
- на ред 12 добавяме реброто $(q, -\log \varphi(q), (q, f(q)))$ от E_1 (на този ред се добавят всички ребра от E_1);
- на ред 13 добавяме новия връх $(q, f(q))$ (на този ред се добавят всички върхове, които не са състояния на изходния f-преобразувател);
- на ред 17 добавяме ребро $((q, f(q)), -\log \lambda(f(q), a), \delta(f(q), a))$ от E_2 за всяко a, за което $(q, a) \notin \mathrm{Dom}(\delta) \wedge (f(q), a) \in \mathrm{Dom}(\delta)$ (на този ред се добавят всички ребра от E_2);
- на ред 19 добавяме ребро $((q, f(q)), -\log \varphi(f(q)), (f(q), f(f(q))))$ от E_3, ако $!\, f(f(q))$, тоест $f[p] \neq 0$ от ред 18 (на този ред се добавят всички ребра от E_3).

Можем да забележим, че тъй като състоянията на f-преобразувателя се разглеждат от **for** цикъла от ред 7 в топологически ред, при изпълнението на ред 19 стойността $F[p]$ вече ще е била сметната. Алгоритъм 5.4 добавя в E всички ребра на графа $G(\mathcal{F})$, като всяко добавяне

на ребро струва $O(1)$ време. Единствено трябва да съобразим, че **for** цикълът от ред 7 прави не повече от $|f| \in O(|E|)$ итерации, в които не добавя ново ребро, а **for** цикълът от ред 14, прави сумарно не повече от $O(|\delta|) \subseteq O(|E|)$ стъпки, в които не добавя ново ребро в E. Това означава, че времевата сложност на функцията FTransducerGraph е $O(|E| + \sum_{i=1}^{|E|} |E[i]|)$, което знаем, че в случая на f-преобразувател на ниво думи е $O(|f| + |f||\Sigma| \div n)$.

5.5.2 Псевдокод за канонизация на f-преобразуватели на ниво символи

Сега ще представим псевдокод, описващ метода за канонизация на f-преобразуатели на ниво символи. В алгоритъм 5.5 е представена функцията CanonicalSymbolFTransducer, която построява каноничната форма на дадения f-преобразувател $\mathcal{F}$ на ниво символи, като използва и съответния f-преобразувател $\mathcal{F}'$ на ниво думи, от който е построен $\mathcal{F}$. От псевдокода за конструиране на $\mathcal{F}$ от $\mathcal{F}'$, знаем, че състоянията на $\mathcal{F}$ с номера в интервала $[1, |f'|]$ съответстват на състоянията на $\mathcal{F}'$. В първия ред на функцията се пресмятат максималните изходи на състоянията от $\mathcal{F}'$ (по-точно, това са максималните изходи на състоянията на хомоморфния образ на подлежащия преобразувател на $\mathcal{F}'$ относно $\rho_{\log}$). След това на ред 3 се построява графа $G(\mathcal{F})$. На ред 4 в графа, представен чрез E, се добавя нов връх (това е върхът r от конструкцията). В следващия цикъл всички изходящи преходи на състояние i на $\mathcal{F}$, което съответства на състояние на $\mathcal{F}'$, биват заменени с един единствен преход от i към новия връх с номер $|E|$, който има тегло $mso[i]$. Първият елемент от тройката $(0, mso[i], |E|)$ от ред 6 е незначещ, той присъства само и единствено за консистентност с вече добавените преходи в E. След това графът E бива обърнат и върху новополучения граф се смятат по стандартен начин с динамично програмиране теглата на най-леките пътища от състоянието $|E|$ до всяко от останалите състояния. Функцията AcyclicShortestPaths прави точно това за линейно време относно големината на входния граф. Можем да забележим, че тъй като всеки връх с номер $1 \leq i \leq |f'|$ има единствен изходящ преход (този преход влиза в $|E|$ и има тегло $mso[i]$), то $mso'[i] = mso[i]$. Останалите редове във функцията CanonicalSymbolFTransducer са същите като в CanonicalWordFTransducer и следват конструкцията за канонична форма на f-преобразувател. Времевите сложности на съставните части на алгоритъм 5.5 са

- $O(|f'||\Sigma'| + |f'| \log(|f'|))$ или $O(|f'||\Sigma'| \div n + |f'| \log(|f'|))$ за ред 2;
- $O(|f| + |f||\Sigma| \div n)$ за ред 3;
- $O(|f'|)$ за редовете от 4 до 6;
- $O(|f| + |f||\Sigma| \div n)$ за ред 7;
- $O(|\lambda|)$ за ред 8;
- $O(|\lambda|)$ за **for** цикъла от ред 9;
- $O(|f|)$ за ред 12;
- $O(|f|)$ за **for** цикъла от ред 13,

където Σ' е входната азбука на $\mathcal{F}'$, Σ е входната азбука на $\mathcal{F}$ и n е реда на езиковия модел, представен чрез $\mathcal{F}$ и $\mathcal{F}'$. Така, тъй като знаем, че в случая на f-преобразувател на ниво символи, $|\lambda| \in O(|f|)$, получаваме, че времевата сложност на CannonicalFTransducer е

$$O\left(|f'||\Sigma'| + |f'| \log(|f'|) + |f| + |f||\Sigma| \div n\right),$$

ако f-преобразувателят на ниво думи има преходи с изходи по-големи от 1, и

$$O\left(|f'||\Sigma'| \div n + |f'|\log(|f'|) + |f| + |f||\Sigma| \div n\right),$$

в противен случай.

Алгоритъм 5.5. Построява каноничната форма на дадения f-преобразувател на ниво символи, като използва и съответния f-преобразувателя на ниво думи.

```
 1: function CanonicalSymbolFTransducer((s', δ', f', λ', φ'), (s, δ, f, λ, φ))
 2:     mso ← WordFTransducerMso((s', δ', f', λ', φ'))
 3:     E ← FTransducerGraph((s, δ, f, λ, φ))
 4:     AppendArray(E, NewArray())
 5:     for i ← 1, |f'| do
 6:         E[i] ← NewArray(1, (0, mso[i], |E|))
 7:     mso' ← AcyclicShortestPaths(|E|, ReverseGraph(E))
 8:     λ'' ← λ
 9:     for all (q, a, r) ∈ λ'' do
10:         r' ← r + mso'[HashSearch(δ, (q, a))] − mso'[q]
11:         HashInsert(λ'', (q, a), exp(−r'))
12:     φ'' ← φ
13:     for all i ← 1, |f| do
14:         r' ← φ[i] + mso'[f[i]] − mso'[i]
15:         φ''[i] ← exp(−r')
16:     return (s, δ, f, λ'', φ'')
```

6 Минимизация на автомати, преобразуватели и f-преобразуватели

В тази глава ще дефинираме понятието „псевдоминимален f-преобразувател" – f-преобразувател, чийто подлежащ преобразувател е минимален. Това понятие е различно от понятието „минимален f-преобразувател" – f-преобразувател с минимален размер (брой състояния плюс брой преходи). Въпреки това всеки псевдоминимален f-преобразувател, еквивалентен на даден изходен f-преобразувател $\mathcal{F}$, има не повече състояния и преходи от $\mathcal{F}$. Тоест можем да намалим размера на даден f-преобразувател, като го пседвоминимизираме. В рамките на тази глава, позовавайки се на класическата теория за минимизация на крайни автомати (Hopcroft и Ullman 1979) и теорията за минимизация на крайни преобразуватели (Mihov и Schulz, 2018 (in press), Mohri 1997 и Mohri 2000), ще покажем как можем да псевдоминимизираме f-преобразувателите, които успяхме да канонизираме в предходната глава. Накрая ще покажем по-ефективен (от гледна точна на пространствена сложност) начин за псевдоминимизация на f-преобразуватели, представящи езикови модели.

6.1 Минимизация на автомати

Първо, ще представим накратко някои резултати от класическата теория на крайните автомати (Hopcroft и Ullman 1979), свързани с тяхната минимизация.

Определение 6.1. *Краен детерминиран автомат* е петорка $(\Sigma, Q, s, F, \delta)$, където

- Σ е входна азбука;
- Q е крайно множество от състояния;
- $s \in Q$ е начално състояние;
- $F \subseteq Q$ е множество от финални състояния;
- $\delta \colon Q \times \Sigma \to Q$ е частична функция на преходите.

Определение 6.2. Нека $(\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. Функцията $\delta^\star \colon Q \times \Sigma^\star \to Q$ дефинираме индуктивно

- $\delta^\star(q, \varepsilon) = q$ за всяко $q \in Q$;
- $\delta^\star(q, \alpha a) = \delta(\delta^\star(q, \alpha), a)$ за всяко $q \in Q$, $\alpha \in \Sigma^\star$ и $a \in \Sigma$.

Определение 6.3. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. *Език на състоянието* $q \in Q$ наричаме множеството

$$\mathcal{L}_{\mathcal{A}}(q) = \{\alpha \in \Sigma^\star \mid \delta^\star(q, \alpha) \in F\}.$$

Определение 6.4. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. *Език на* $\mathcal{A}$ наричаме множеството $\mathcal{L}_{\mathcal{A}}(s)$, което означаваме с $\mathcal{L}(\mathcal{A})$.

Определение 6.5. Нека $\mathcal{A}$ и $\mathcal{A}'$ са крайни детерминирани автомати. Казваме, че $\mathcal{A}$ е *еквивалентен на* $\mathcal{A}'$, ако $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

Определение 6.6. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. Казваме, че $\mathcal{A}$ е *минимален автомат за* $\mathcal{L}(\mathcal{A})$, ако за всеки краен детерминиран автомат $\mathcal{A}' = (\Sigma, Q', s', F', \delta')$ е в сила, че

$$\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A}) \implies |Q| \leq |Q'|.$$

В този раздел под „минимизация на краен детерминиран автомат" ще имаме предвид построяване на минимален автомат за езика на изходния автомат (тоест еквивалентен на изходния автомат). Минималните автомати, еквивалентни на краен детерминиран автомат, могат да бъдат характеризирани със свойството, че всичките им състояния са достижими и кодостижими и нямат различни еквивалентни състояния.

Определение 6.7. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. Казваме, че състоянието $q \in Q$ е

- *достижимо*, ако $(\exists \alpha \in \Sigma^\star)(\delta^\star(s, \alpha) = q)$;
- *кодостижимо*, ако $(\exists \alpha \in \Sigma^\star)(\delta^\star(q, \alpha) \in F)$.

Определение 6.8. Нека $\mathcal{A}$ е краен детерминиран автомат. Казваме, че $\mathcal{A}$ е *тримован*, ако всяко негово състояние е достижимо и кодостижимо.

Определение 6.9. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е краен детерминиран автомат. Релацията $\equiv_{\mathcal{A}} \subseteq Q \times Q$ дефинираме за $p, q \in Q$ така

$$p \equiv_{\mathcal{A}} q \iff \mathcal{L}_{\mathcal{A}}(p) = \mathcal{L}_{\mathcal{A}}(q).$$

Състояния p и q, такива че $p \equiv_{\mathcal{A}} q$, наричаме *еквивалентни*.

Твърдение 6.1. *Нека* $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ *е краен детерминиран автомат. Тогава* $\mathcal{A}$ *е минимален автомат за* $\mathcal{L}(\mathcal{A})$*, тогава и само тогава, когато* $\mathcal{A}$ *е тримован и*

$$(\forall p \in Q)(\forall q \in Q)(p \neq q \implies p \not\equiv_{\mathcal{A}} q). \qquad \square$$

Изискването автоматът да бъде тримован е несъществено, тъй като за всеки краен детерминиран автомат $\mathcal{A}$ съществува тримован краен детерминиран автомат $\mathcal{A}'$, такъв че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$, който може да бъде построен за линейно време относно големината на $\mathcal{A}$. Затова до края на този раздел ще считаме, че имаме тримован краен детерминиран автомат. Тъй като релацията $\equiv_{\mathcal{A}}$ е релация на еквивалентност и всеки две еквивалентни състояния могат да бъдат заменени с едно състояние без да бъде променен езикът на автомата, минимизацията на краен детерминиран автомат $\mathcal{A}$ се свежда до намирането на класовете на еквивалентност на $\equiv_{\mathcal{A}}$.

Твърдение 6.2. *Нека* $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ *е тримован краен детерминиран автомат. Тогава крайният детерминиран автомат* $\mathcal{A}' = (\Sigma, Q', s', F', \delta')$*, където*

- $Q' = \{[q]_{\equiv_\mathcal{A}} \mid q \in Q\}$;
- $s' = [s]_{\equiv_\mathcal{A}}$;
- $F' = \{[q]_{\equiv_\mathcal{A}} \mid q \in F\}$;
- $\delta'([q]_{\equiv_\mathcal{A}}, a) = [\delta(q,a)]_{\equiv_\mathcal{A}}$ *за всяко* $q \in Q$ *и* $a \in \Sigma$,

е минимален автомат за $\mathcal{L}(\mathcal{A})$. □

Алгоритъмът за минимизация на краен автомат, който следва да разгледаме, построява минималния автомат от предходното твърдение, като се възползва съществено от свойството, че еквивалентността на две състояния на краен детерминиран автомат се свежда до еднаквостта на тяхната финалност и еквивалентността на техните директни наследници (тоест достижимите от тях с един преход състояния).

Твърдение 6.3. *Нека* $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ *е тримован краен детерминиран автомат. Тогава за всеки две състояния* $p, q \in Q$ *е в сила, че*

$$p \equiv_\mathcal{A} q \iff [(p \in F \iff q \in F) \wedge (\forall a \in \Sigma)(\delta(p,a) \equiv_\mathcal{A} \delta(q,a))].$$ □

6.1.1 Алгоритъм на Béal и Crochemore

Алгоритъмът на Béal и Crochemore (Béal и Crochemore 2008), който всъщност е вариант на алгоритъма на Hopcroft (Hopcroft 1971), пресмята по даден тримован краен детерминиран автомат $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ класовете на еквивалентност на $\equiv_\mathcal{A}$, като постъпково изфинява релация, която е не по-фина от $\equiv_\mathcal{A}$. Както вече споменахме, ограничението изходният автомат да бъде тримован е несъществено. Алгоритъмът започва със съгласувано с $\mathcal{A}$ разбиване на Q и тестово множество за това разбиване.

Определение 6.10. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е тримован краен детерминиран автомат. Казваме, че $\{B_i\}_{i=1}^m$ е *съгласувано с* $\mathcal{A}$ *разбиване на* Q, ако

- $(\forall 1 \leq i \leq m)(B_i \neq \varnothing)$;
- $(\forall 1 \leq i, j \leq m)(i \neq j \implies B_i \cap B_j = \varnothing)$;
- $\bigcup_{i=1}^m B_i = Q$;
- $(\forall 1 \leq i, j \leq m)(\forall p \in B_i)(\forall q \in B_j)(i \neq j \implies p \not\equiv_\mathcal{A} q)$.

Ясно е, че всяко съгласувано с $\mathcal{A}$ разбиване на Q представлява не по-фина от $\equiv_\mathcal{A}$ релация.

Определение 6.11. Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е тримован краен детерминиран автомат и $\{B_i\}_{i=1}^m$ е съгласувано с $\mathcal{A}$ разбиване на Q. Казваме, че T е *тестово множество* за $\{B_i\}_{i=1}^m$, ако за всяко $1 \leq i \leq m$ и всеки $p, q \in B_i$, за които $p \not\equiv_\mathcal{A} q$, съществува $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \ldots a_n$, такова че

- $(\forall 0 \leq j < n)(\exists 1 \leq k \leq m)(\delta^\star(p, a_1 a_2 \ldots a_j) \in B_k \wedge \delta^\star(q, a_1 a_2 \ldots a_j) \in B_k)$;
- $(\exists k \in T)(\delta^\star(q,\alpha) \in B_k \iff \delta^\star(q,\alpha) \notin B_k)$.

Ако имаме съгласувано с $\mathcal{A}$ разбиване на Q, което разграничава финалните от нефиналните състояния, елементите на всяко тестово множество за това разбиване представляват „свидетели" за нееквивалентността на всички двойки състояния, които са от една и съща група в разбиването и не са еквивалентни. Ако постигнем съгласувано с $\mathcal{A}$ разбиване на Q, разграничаващо финалните от нефиналните състояния, и празно тестово множество за това разбиване, то елементите на разбиването ще бъдат точно класовете на еквивалетност на $\equiv_{\mathcal{A}}$

Лесно се съобразява, че $\{F, Q\setminus F\}\setminus\{\varnothing\}$ е съгласувано с $\mathcal{A}$ разбиване на Q и $\{F, Q\setminus F\}\setminus\{\varnothing\}$ е тестово множество за $\{F, Q \setminus F\} \setminus \{\varnothing\}$. Алгоритъмът на Béal и Crochemore започва с това разбиване и тестово множество. Целта на алгоритъма е на всяка стъпка да изфинява съгласуваното с $\mathcal{A}$ разбиване на Q, използвайки някой от свидетелите в тестовото множество. Инвариантът, който се поддържа е, че на всяка стъпка имаме съгласувано с изходния автомат разбиване на състоянията и тестово множество за това разбиване. Следващото твърдение показва стратегията за изфиняване, която се използва в алгоритъма.

Твърдение 6.4. *Нека $\Sigma = \{a_1, a_2, \ldots, a_n\}$ е азбука, $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е тримован краен детерминиран автомат, $\mathcal{B} = \{B_i\}_{i=1}^m$ е съгласувано с $\mathcal{A}$ разбиване на Q и T е тестово множество за $\mathcal{B}$. Нека $B \in T$ и за $1 \le i \le k$ и $s_1 s_2 \ldots s_n \in \{0,1\}^n$*

$$B_i^{s_1 s_2 \ldots s_n} = \{q \in B_i \mid (\forall 1 \le j \le n)(\delta(q, a_j) \in B \iff s_j = 1)\}.$$

Тогава

- *B' е съгласувано с $\mathcal{A}$ разбиване на Q, където*

$$B' = \bigcup_{i=1}^{k} B'_i,$$
$$B'_i = \{B_i^{s_1 s_2 \ldots s_n} \mid s_1 s_2 \ldots s_n \in \{0,1\}^n, B_i^{s_1 s_2 \ldots s_n} \neq \varnothing\},\ \text{за } 1 \le i \le k;$$

- *T' е тестово множество за B', където*

$$T' = \left(\bigcup_{\substack{1 \le i \le m \\ B_i \in T\setminus\{B\}}} B'_i \right) \cup \left(\bigcup_{\substack{1 \le i \le m \\ B_i \notin T\setminus\{B\} \\ |B'_i| > 1}} B'_i \setminus \operatorname*{argmax}_{X \in B'_i} |X| \right).$$ □

Алгоритъмът завършва (достига до празно тестово множество), тъй като всяко състояние $q \in Q$ се „разглежда" краен брой (най-много $\log_2 |Q|$) пъти. Казваме, че състояние $q \in Q$ *се разглежда* в момента t, ако в момента t избираме свидетеля $B \supseteq \{q\}$ от тестовото множество и изфиняваме съгласуваното с $\mathcal{A}$ разбиване на Q съгласно B.

Твърдение 6.5. *Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е тримован краен детерминиран автомат. Нека $t_1 < t_2$ и $\mathcal{B}^{(t_1)}$ ($\mathcal{B}^{(t_2)}$) и $T^{(t_1)}$ ($T^{(t_2)}$) са съответно съгласуваното с $\mathcal{A}$ разбиване на Q и тестовото множество за $\mathcal{B}^{(t_1)}$ ($\mathcal{B}^{(t_2)}$) в t_1-вия (t_2-рия) момент от изпълнението на алгоритъма на Béal и Crochemore върху $\mathcal{A}$. Нека състоянието $q \in Q$ се разглежда в моментите t_1 и t_2, тоест $q \in B'$, $B' \in \mathcal{B}^{(t_1)}$ и $q \in B''$, $B'' \in \mathcal{B}^{(t_2)}$. Тогава*

$$|B''| \le \frac{1}{2}|B'|.$$ □

Твърдение 6.6. *Нека $\mathcal{A} = (\Sigma, Q, s, F, \delta)$ е тримован краен детерминиран автомат. Всяко състояние $q \in Q$ се разглежда най-много $\log_2 |Q|$ пъти по време на изпълнението на алгоритъма на Béal и Crochemore върху $\mathcal{A}$.* □

От предходното твърдение и фактът, че всяко разглеждане на състояние $q \in Q$ може да се имплементира за линейно време относно броя на влизащите в q преходи, следва, че алгоритъмът може да се имплементира за време $O(|\Sigma| + |\operatorname{Dom}(\delta)| \log_2 |Q|)$ и памет $O(|\Sigma| + |\operatorname{Dom}(\delta)|)$. Повече подробности за времевата и пространствената сложност на алгоритъма са изложени в Béal и Crochemore 2008.

6.1.2 Псевдокод за алгоритъма на Béal и Crochemore

В този подраздел ще дадем описание на алгоритъма на Béal и Crochemore чрез псевдокод, който постига горепосочените сложности по време и памет. Обосновка за неговата коректност, както и подробни коментари по имплементацията, няма да даваме, тъй като те могат да бъдат намерени в Hopcroft 1971 и Béal и Crochemore 2008.

Алгоритъм 6.1. Построява минимален автомат за $\mathcal{L}(\mathcal{A})$ по даден тримован краен детерминиран автомат $\mathcal{A}$.

```
1:  function BéalCrochemoreMinimization((s, F, δ))
2:      (B, I) ← StateEquivalence(F, δ)
3:      s' ← 0
4:      F' ← NewArray(|B|, 0)
5:      δ' ← NewArray(|B|, NewArray())
6:      for i ← 1, |B| do
7:          for all q ∈ B[i] do
8:              if q = s then
9:                  s' ← i
10:         q ← HeadList(B[i])
11:         F'[i] ← F[q]
12:         for all (a, p) ∈ δ[q] do
13:             ArrayAppend(δ'[i], (a, I[p]))
14:     return (s', F', δ')
```

Функцията BéalCrochemoreMinimization в алгоритъм 6.1 построява по даден тримован краен детерминиран автомат $\mathcal{A}$ минимален автомат за $\mathcal{L}(\mathcal{A})$. Първо на ред 2 с алгоритъма на Béal и Crochemore се намират класовете на еквивалентност на релацията $\equiv_{\mathcal{A}}$. Те са представени посредством масивите $\mathcal{B}$ и $\mathcal{I}$. Всеки елемент $\mathcal{B}[i]$ за $1 \leq i \leq |\mathcal{B}|$ е списък от състояния и представлява един клас на еквивалентност на релацията $\equiv_{\mathcal{A}}$. Масивът $\mathcal{I}$ е с дължина $|F|$ и $\mathcal{I}[i]$ за $1 \leq i \leq |F|$ представлява индекса на списъка в $\mathcal{B}$, който съдържа състоянието с номер i. На редове 3, 4 и 5 се създава представянето на резултатния минимален автомат за $\mathcal{L}(\mathcal{A})$. То се състои от номера на началното състояние s', масив F' с дължина $|\mathcal{B}|$ (колкото ще бъдат състоянията на минималния автомат), такъв че $F'[i]$ за $1 \leq i \leq |\mathcal{B}|$ е 1, ако състоянието с номер i е финално, и 0 в противен случай, и масив δ', който за всяко състояние $1 \leq i \leq |\mathcal{B}|$ ще пази в $\delta'[i]$ масив, съдържащ всички изходящи преходи на състоянието i. Построяването на минималния автомат представлява обхождане на всички негови състояния $1 \leq i \leq |\mathcal{B}|$. За състоянието с номер i **for** цикълът на ред 7 проверява дали то е началното състояние на резултатния автомат, ред 11 проверява финалността на състоянието (то е финално тогава и само тогава, когато има финално състояние в този

клас на еквивалентност) и **for** цикълът на ред 12 добавя в резултатния автомат изходящите преходи на състоянието (тъй като всичките състояния в този клас на еквивалентност са еквивалентни, всяко от тях има еднакви изходящи преходи). Алгоритъм 6.1 директно имплементира конструкцията от твърдение 6.2. Считайки, че функцията StateEquivalence може да се имплементира за време $O(|\Sigma| + |\operatorname{Dom}(\delta)| \log |Q|)$ и памет $O(|\Sigma| + |\operatorname{Dom}(\delta)|)$, времевата сложност на BéalCrochemoreMinimization е $O(|\Sigma| + |\operatorname{Dom}(\delta)| \log |Q|)$, а пространствената ѝ сложност е $O(|\Sigma| + |\operatorname{Dom}(\delta)|)$, тъй като останалите редове на фунцията отнемат не повече от линейно време и памет относно големината на изходния автомат.

Алгоритъм 6.2. Пресмята класовете на еквивалентност на релацията $\equiv_{\mathcal{A}}$, където $\mathcal{A}$ е дадения тримован краен детерминиран автомат.

```
1:  function StateEquivalence(F, δ)
2:      (B, I, P, T, Y) ← InitialPartition(F)
3:      Σ' ← NewArray(AlphabetSize(δ), NewList())
4:      Σ'' ← NewArray(AlphabetSize(δ), NewList())
5:      σ ← NewArray(|F|, NewArray())
6:      τ ← NewArray(|B|, NewList())
7:      δ^{-1} ← ReverseGraph(δ)
8:      while |T| ≠ 0 do
9:          t ← ListHead(T)
10:         ListRemove(T, t)
11:         Y[t] ← 0
12:         RefinePartition((B, I, P, T, Y), (Σ', Σ'', σ, τ), δ^{-1}, B[t])
13:     return (B, I)
```

Функцията StateEquivalence в алгоритъм 6.2 пресмята класовете на еквивалентност на релацията $\equiv_{\mathcal{A}}$ за тримован краен детерминиран автомат $\mathcal{A}$, посредством описания метод на Béal и Crochemore. На ред 2 се пресмята началното разбиване на състоянията. То се представя чрез масивите $\mathcal{B}, \mathcal{I}, \mathcal{P}$ и $\mathcal{Y}$ и списъка $\mathcal{T}$. Вече обяснихме семантиката на масивите $\mathcal{B}$ и $\mathcal{I}$. Масивът $\mathcal{P}$ е с дължина $|F|$ и $\mathcal{P}[i]$ за $1 \leq i \leq |F|$ представлява указател към мястото в $\mathcal{B}$, където е записано състоянието i. Целта на $\mathcal{P}$ е да позволява бързо (за константно време) премахване на дадено състояние от неговата текуща група в $\mathcal{B}$. $\mathcal{T}$ е списък от числа. Всяко число в $\mathcal{T}$ е индекс на клас на еквивалентност от $\mathcal{B}$. Така $\mathcal{T}$ представя текущото тестово множество за разбиването $\mathcal{B}$. Масивът $\mathcal{Y}$ е с дължина $|\mathcal{B}|$ и $\mathcal{Y}[i]$ за $1 \leq i \leq |\mathcal{B}|$ е 1, ако числото i е в тестовото множество $\mathcal{T}$, и 0 в противен случай. Редовете от 3 до 6 дефинират помощни структури от данни, които се използват от функцията, която изфинява текущото разбиване. Функцията AlphabetSize намира големината на азбуката на автомата от неговата δ-функция, като намира максималната буква, с която в автомата има преход (считаме, че азбуката е представена чрез числа в интервала $[1, |\Sigma|]$). Ред 7 обръща посоката на преходите в δ, така че да можем бързо да намираме всички преходи влизащи в дадено състояние. Останалите редове от функцията следват гореописаната стратегия за изфиняване на разбиването посредством кой да е свидетел от тествовото множество. По-долу ще покажем, че функцията InitialPartition отнема $O(|F|)$ време. Функцията ReverseGraph отнема $O(|\operatorname{Dom}(\delta)|)$ време. Тоест редовете от 2 до 7 отнемат $O(|\Sigma| + |\operatorname{Dom}(\delta)|)$ време и $O(|\Sigma| + |\operatorname{Dom}(\delta)|)$ памет. Знаем че всяко състояние ще се разгледа най-много $\log(|Q|)$ пъти. По-долу ще покажем, че времевата сложност на процедурата RefinePartition е линейна относно броя преходи, които влизат в състояния от свидетеля, спрямо който се изфинява текущото разбиване. Това означава, че времето, което отнема разглеждането

на едно състояние е линейно относно броя на влизащите в него преходи. Тоест времевата сложност на **while** цикъла в алгоритъм 6.2 е

$$O(\sum_{q\in Q} \log|Q| \sum_{(p,a,q)\in\delta} 1) = O(\log|Q| \sum_{q\in Q} \sum_{(p,a,q)\in\delta} 1) = O(|\operatorname{Dom}(\delta)| \log|Q|).$$

Така получаваме точно обещаните сложности по време и памет.

Алгоритъм 6.3. Построява началното разбиване и тестово множество от алгоритъма на Béal и Crochemore.

```
1:  function InitialPartition(F)
2:      B_1 ← NewList()
3:      B_2 ← NewList()
4:      I ← NewArray(|F|, 0)
5:      P ← NewArray(|F|, NIL)
6:      for i ← 1, |F| do
7:          if F[i] = 0 then
8:              P[i] ← ListAppend(B_1, i)
9:          else
10:             P[i] ← ListAppend(B_2, i)
11:         I[i] ← F[i] + 1
12:     B ← NewArray()
13:     T ← NewList()
14:     Y ← NewArray()
15:     ListAppend(T, 1)
16:     ArrayAppend(Y, 1)
17:     if |B_1| = 0 then
18:         ArrayAppend(B, B_2)
19:     else
20:         ArrayAppend(B, B_1)
21:         ArrayAppend(B, B_2)
22:         ListAppend(T, 2)
23:         ArrayAppend(Y, 1)
24:     return (B, I, P, T, Y)
```

Функцията InitialPartition в алгоритъм 6.3 пресмята началното съгласувано разбиване на състоянията и тестовото множество за това разбиване от алгоритъма на Béal и Crochemore. Няма да разглеждаме в подобности тази функция, тъй като вече обяснихме семантиката на структурите от данни, които построява, и стъпките, които предприема алгоритъма са достатъчно прости. Единствената част от алгоритъм 6.3, която не е с константна времева сложност е **for** цикълът от ред 6, който очевидно прави $|F|$ константни итерации. Затова времевата сложност на функцията InitialPartition е $O(|F|) = O(|Q|)$.

Фунцкията RefinePartition в алгоритъм 6.4 изфинява текущото съгласувано разбиване на състоянията съгласно даден свидетел B от тестовото множество за това разбиване. Разбиването и тестовото множество са представени чрез структурите от данни $(\mathcal{B}, \mathcal{I}, \mathcal{P}, \mathcal{T}, \mathcal{Y})$. В процеса на изпълнение функцията използва помощните структури от данни $(\Sigma', \Sigma'', \sigma, \tau)$

Първата стъпка на алгоритъма (редове 2-7) е да запише в списъка Σ'_i всички букви, такива че има влизащи преходи с тези букви в състояния от свидетеля. В тази стъпка всички състояния, които имат влизащ преход в състояние от свидетеля B, се разпределят в

Алгоритъм 6.4. Изфинява текущото разбиване съгласно свидетеля B от тестовото множество.

```
1: procedure RefinePartition((B, I, P, T, Y), (Σ', Σ'', σ, τ), δ^{-1}, B)
2:     Σ'_i ← NewList()
3:     for all q ∈ B do
4:         for all (a, p) ∈ δ^{-1}(q) do
5:             if |Σ'[a]| = 0 then
6:                 ListAppend(Σ'_i, a)
7:             ListAppend(Σ'[a], p)
```

списъците от Σ', спрямо това дали имат входящ преход с тази буква или не. Това става за линейно време относно броя на преходите, влизащи в състояния от B. Целта на тази стъпка е да се фиксира наредба на символите от азбуката.

```
8:     σ_i ← NewList()
9:     for all a ∈ Σ'_i do
10:        for all q ∈ Σ'[a] do
11:            if |σ[q]| = 0 then
12:                ListAppend(σ_i, q)
13:            ArrayAppend(σ[q], a)
14:        ListEmpty(Σ'[a])
15:    ListEmpty(Σ'_i)
```

В следващата стъпка на алгоритъма (редове 8-15) се намира списъка σ_i от всички състояния, които имат входящ преход в състояние от B. За всяко състояние q от списъка σ_i в $\sigma[q]$ се записват всички уникални символи, с които то има преход в състояние от свидетеля, като символите са сортирани относно фиксираната наредба в Σ'_i. Масива $\sigma[q]$ наричаме сигнатура на състоянието q. Тази стъпка отнема не повече време от броя на преходите, влизащи в състояние от B.

```
16:    for all q ∈ σ_i do
17:        if |Σ'[|σ[q]|]| = 0 then
18:            ListAppend(Σ'_i, |σ[q]|)
19:        ListAppend(Σ'[|σ[q]|], q)
20:    ListEmpty(σ_i)
```

В редовете от 16 до 20 състоянията от σ'_i се разпределят в списъците от Σ' съгласно дължините на техните сигнатури. В Σ'_i се пазят всички различни дължини на сигнатури. Тази стъпка се извършва за същото време като предишната.

```
21:    Σ''_i ← NewList()
22:    τ_i ← NewList()
23:    for all n ∈ Σ'_i do
```

В редовете от 21 до 23 започва цикъл, чиято цел е да сортира състоянията съгласно –

първо дължините на техните сигнатури, след това техните групи в разбиването и накрая съгласно самите сигнатури.

```
24:  for all q ∈ Σ'[n] do
25:      if |τ[ℐ[q]]| = 0 then
26:          ListAppend(τ_i, ℐ[q])
27:      ListAppend(τ[ℐ[q]], q)
28:  ListEmpty(Σ'[n])
```

В редовете от 24 до 28 в списъка τ_i се записват номерата на всички групи от разбиването, които съдържат състояние, което има сигнатура с дължина n и което има преход към състояние от B. В списъка $\tau[i]$ за $1 \leq i \leq |\mathcal{B}|$ се записват всички състояния от $\Sigma'[n]$, които принадлежат на $\mathcal{B}[i]$. Това става за линейно време относно броя състояния с тази дължина на сигнатура.

```
29:  for all t ∈ τ_i do
30:      for all q ∈ τ[t] do
31:          ListAppend(Σ'[n], q)
32:      ListEmpty(τ[t])
33:  ListEmpty(τ_i)
```

В редовете от 29 до 33 се сортират състоянията с дължина на сигнатурите равна на n съгласно номерата на техните групи в разбиването. Това става за линейно време относно броя състояния с тази дължина на сигнатура.

```
34:  for i ← 1, n do
35:      for all q ∈ Σ'[n] do
36:          if |Σ''[σ[q][i]]| = 0 then
37:              ListAppend(Σ''_i, σ[q][i])
38:          ListAppend(Σ''[σ[q][i]], q)
39:      ListEmpty(Σ'[n])
40:      for all a ∈ Σ''_i do
41:          for all q ∈ Σ''[a] do
42:              ListAppend(Σ'[n], q)
43:          ListEmpty(Σ''[a])
44:      ListEmpty(Σ''_i)
```

В редовете от 34 до 44 се прилага Bucket Sort за всяка координата на сигнатурите на състоянията от $\Sigma'[n]$. Така накрая $\Sigma'[n]$ представлява списък от същите състояния, сортиран по групите на състоянията в разбиването и по техните сигнатури. Тази стъпка отнема линейно време относно броя на входящите преходи на състоянията, които биват сортирани.

```
45:  for all q ∈ Σ'[n] do
46:      ListAppend(σ_i, q)
47:  ListEmpty(Σ'[n])
```

В редовете от 45 до 47 списъците от състояния $\Sigma'[n]$ за $n \in \Sigma'_i$ биват обединени в списъка σ_i. Така получаваме подредба на състоянията с входящи преходи в B по дължина на сигнатурите, по групата от разбиването, на която принадлежат, и накрая по самата сигнатура. Състоянията, които са неразличими относно гореспоменатите характеристики, ще бъдат последователни в подредбата.

```
48:  ListEmpty(Σ')
49:  prev ← 0
50:  B_new ← NewList()
51:  for all q ∈ σ_i do
52:      if prev ≠ 0 and (I[prev] ≠ I[q] or σ[prev] ≠ σ[q]) then
53:          if τ[I[prev]] = 0 then
54:              ListAppend(τ_i, I[prev])
55:          ListAppend(τ[I[prev]], B_new)
56:          B_new ← NewList()
57:      ListRemove(B[I[q]], P[q])
58:      P[q] ← ListAppend(B_new, q)
59:      prev ← q
60:  ListRemove(B[I[prev]], P[prev])
61:  P[prev] ← ListAppend(B_new, prev)
62:  ListEmpty(σ_i)
63:  for all j ∈ τ_i do
64:      M ← ListHead(τ[j])
65:      for all X ∈ τ[j] do
66:          if |X| > |M| then
67:              M ← X
68:      if |B[j]| = 0 then
69:          B[j] ← M
70:          ListRemove(τ[j], M)
71:      else if |B[j]| < |M| then
72:          M' ← M
73:          M ← B[j]
74:          B[j] ← M'
75:      for all q ∈ B[j] do
76:          I[q] ← j
77:      for all X ∈ τ[j] do
78:          ListAppend(B, X)
79:          ListAppend(T, |B|)
80:          ArrayAppend(Y, 1)
81:          for all q ∈ X do
82:              I[q] ← |B|
83:      ListEmpty(τ[j])
84:  ListEmpty(τ_i)
```

В редовете от 51 до 61 състоянията от σ_i биват групирани съгласно техните характеристики. Новополучените групи биват групирани в τ спрямо номера на класа на еквивалентност на състоянията в групата. В τ_i се записват всички номера на класове на еквивалентност,

които съдържат състояние от σ_i. Междувременно се обновяват указателите в масива $\mathcal{P}$.

Всяка итерация на цикъла от ред 63 обновява $(\mathcal{B}, \mathcal{I}, \mathcal{P}, \mathcal{T}, \mathcal{Y})$, като разбива класа на еквивалентност с номер j. В редовете от 64 до 67 се намира най-голямата група от $\sigma[j]$ (тези групи заедно евентуално с $\mathcal{B}[j]$ представляват разбиване на класа на еквивалентност с номер j). В редовете от 68 до 74 максималната група от разбиването на j-тия клас бива записана в $\mathcal{B}[j]$. **for** цикълът на ред 75 обновява стойностите $\mathcal{I}[q]$ за всяко състояние q от $\mathcal{B}[j]$. Последният цикъл записва останалите групи в $\mathcal{B}$ и $\mathcal{T}$. Тоест в $\mathcal{B}$ влизат всички групи от разбиването на класа на еквивалентност j, а в $\mathcal{T}$ всички освен най-голямата група (ако $\mathcal{T}$ съдържа j, най-голямата група също ще влезе в $\mathcal{T}$, както гласи стратегията на алгоритъма). Това означава, че ако j-тият клас не се разбие в $\mathcal{T}$ няма да бъде добавено нищо (ако j е било в $\mathcal{T}$, то в този случай j ще остане в $\mathcal{T}$). Структурите от данни $\mathcal{I}$ и $\mathcal{Y}$ биват обновени, така че да отразяват промените в $\mathcal{B}$ и $\mathcal{T}$. Очевидно е, че тази част от алгоритъма отнема $O(|\sigma_i|)$ време, което заедно с останалите стъпки означава, че процедурата RefinePartition има времева сложност $O(|\sigma_i|)$, тоест линейно време относно броя на преходите, които влизат в свидетеля B.

6.2 Минимизация на канонични преобразуватели

В този раздел, следвайки подхода от предходния, ще представим накратко някои резултати от теорията на крайните преобразуватели (Mihov и Schulz, 2018 (in press), Mohri 1997 и Mohri 2000), свързани с минимизация на преобразуватели. В тези източници се показва как могат да бъдат минимизирани преобразуватели с изходи от моноида $(\Sigma^\star, \cdot, \varepsilon)$, където $\cdot$ е операцията „конкатенация на думи", а в Mohri 1997 се разказва и за минимизация на преобразуватели с изходи от $\mathcal{R}_{[0,1]}$. Същото, обаче, може да се покаже и за преобразуватели с изходи от $\mathcal{R}_{\geq 0}$, чиито максимални изходи съществуват.

Минимизацията на един преобразувател изисква първо той да бъде канонизиран. От предходния раздел знаем, че преобразувателите с изходи от $\mathcal{R}_{[0,1]}$ могат да бъдат канонизирани. Също така показахме, че преобразувателите с изходи от $\mathcal{R}_{\geq 0}$, които са подлежащи преобразуватели на f-преобразуватели на ниво символи, могат да бъдат канонизирани. Вярно е дори по-общо твърдение – преобразувателите $\mathcal{T}$ с изходи от $\mathcal{R}_{\geq 0}$, за които $\mathrm{mso}_{\mathcal{T}}(q)$ е дефинирано за всяко състояние q на $\mathcal{T}$, могат да бъдат канонизирани. [1] Затова под каноничен преобразувател в тази глава ще разбираме преобразувател с изходи от $\mathcal{R}_{\geq 0}$, чиито максимални изходи са дефинирани и равни на 1.

Определение 6.12. Нека $\mathcal{T}$ и $\mathcal{T}'$ са преобразуватели. Казваме, че $\mathcal{T}$ е *еквивалентен на* $\mathcal{T}'$, ако $O_{\mathcal{T}} = O_{\mathcal{T}'}$.

Определение 6.13. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Казваме, че преобразувателят $\mathcal{T}$ е *минимален преобразувател за* $O_{\mathcal{T}}$, ако за всеки преобразувател $\mathcal{T}' = (\Sigma, \mathcal{M}, Q', s', F', \delta', \lambda', \iota')$ е в сила, че

$$O_{\mathcal{T}'} = O_{\mathcal{T}} \implies |Q| \leq |Q'|.$$

В този раздел под „минимизация на преобразувател" ще имаме предвид построяване на минимален преобразувател за функцията, представена от изходния преобразувател (тоест еквивалентен на изходния преобразувател). Каноничните минимални преобразуватели могат да бъдат характеризирани със свойството, че всичките им състояния са достижими и

1. Разликата с канонизацията на преобразуватели с изходи от $\mathcal{R}_{[0,1]}$ е, че максималните изходи се намират с алгоритъма на Bellman-Ford (Cormen и др. 2009), а не с алгоритъма на Dijkstra.

кодостижими и нямат различни еквивалентни състояния. Термините „достижимо състояние" и „кодостижимо състояние" на преобразувател дефинираме по същия начин, както дефинирахме тези понятия за състояния на краен детерминиран автомат. По аналогия с определението на тримован краен детерминиран автомат, под тримован преобразувател ще разбираме преобразувател, чиито състояния са достижими и кодостижими.

Определение 6.14. Нека $\mathcal{T} = (\Sigma, \mathcal{M}, Q, s, F, \delta, \lambda, \iota)$ е преобразувател. Релацията $\equiv_{\mathcal{T}} \subseteq Q \times Q$ дефинираме за $p, q \in Q$ така

$$p \equiv_{\mathcal{T}} q \iff O^p_{\mathcal{T}} = O^q_{\mathcal{T}}.$$

Състояния p и q, такива че $p \equiv_{\mathcal{T}} q$, наричаме *еквивалентни*.

Твърдение 6.7. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е каноничен преобразувател. Тогава $\mathcal{T}$ е минимален преобразувател за $O_{\mathcal{T}}$, тогава и само тогава, когато $\mathcal{T}$ е тримован и*

$$(\forall p \in Q)(\forall q \in Q)(p \neq q \implies p \not\equiv_{\mathcal{T}} q). \qquad \square$$

Тук отново изискването преобразувателят да е тримован е несъществено, тъй като за всеки преобразувател $\mathcal{T}$ може да бъде построен тримован преобразувател $\mathcal{T}'$, еквивалентен на $\mathcal{T}$. Аналогично на разсъжденията за релацията $\equiv_{\mathcal{A}}$ за краен детерминиран автомат $\mathcal{A}$, можем да забележим, че $\equiv_{\mathcal{T}}$ е релация на еквивалентност, и да заключим, че минимизацията на даден каноничен преобразувател $\mathcal{T}$ се свежда до намирането на класовете на еквивалентност на $\equiv_{\mathcal{T}}$.

Твърдение 6.8. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е каноничен тримован преобразувател. Тогава преобразувателят $\mathcal{T}' = (\Sigma, \mathcal{R}_{\geq 0}, Q', s', F', \delta', \lambda', \iota)$, където*

- $Q' = \{[q]_{\equiv_{\mathcal{T}}} \mid q \in Q\}$;
- $s' = [s]_{\equiv_{\mathcal{T}}}$;
- $F' = \{[q]_{\equiv_{\mathcal{T}}} \mid q \in F\}$;
- $\delta'([q]_{\equiv_{\mathcal{T}}}, a) = [\delta(q, a)]_{\equiv_{\mathcal{T}}}$ *за всяко* $q \in Q$ *и* $a \in \Sigma$;
- $\lambda'([q]_{\equiv_{\mathcal{T}}}, a) = \lambda(q, a)$ *за всяко* $q \in Q$ *и* $a \in \Sigma$,

е минимален преобразувател за $O_{\mathcal{T}}$. $\square$

В случая на каноничен тримован преобразувател еквивалентността на две състояния се свежда до еднаквостта на тяхната финалност, еднаквостта на изходите на техните преходи и еквивалентността на техните директни наследници.

Твърдение 6.9. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е каноничен тримован преобразувател. Тогава за всеки две състояния $p, q \in Q$ е в сила, че*

$$p \equiv_{\mathcal{T}} q \iff [(p \in F \iff q \in F) \wedge (\forall a \in \Sigma)(\lambda(p, a) = \lambda(q, a) \wedge \delta(p, a) \equiv_{\mathcal{T}} \delta(q, q))]. \qquad \square$$

Използвайки горното твърдение, може да се покаже, че две състояния на каноничен тримован преобразувател са еквивалентни тогава и само тогава, когато същите две състояния са еквивалентни в контекста на тримован краен детерминиран автомат, индуциран от изходния преобразувател.

Твърдение 6.10. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е каноничен тримован преобразувател. Нека азбуката $\Sigma_{\mathcal{A}}$ е дефинирана така*

$$\Sigma_{\mathcal{A}} = \{(a, \lambda(q, a)) \mid (q, a) \in \mathrm{Dom}(\delta)\}$$

и $\mathcal{A} = (\Sigma_{\mathcal{A}}, Q, s, F, \delta_{\mathcal{A}})$ е тримованият краен детерминиран автомат, за който

$$\delta_{\mathcal{A}}(q, (a, b)) = \delta(q, a) \text{ за } q \in Q \text{ и } (a, b) \in \Sigma_{\mathcal{A}}.$$

Тогава за състоянията $p, q \in Q$ е в сила, че

$$p \equiv_{\mathcal{T}} q \iff p \equiv_{\mathcal{A}} q. \qquad \square$$

От предходното твърдение можем заключим, че задачата за минимизация на каноничен тримован преобразувател се свежда до задачата за минимизация на тримован краен детерминиран автомат.

Твърдение 6.11. *Нека $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ е каноничен тримован преобразувател. Нека азбуката $\Sigma_{\mathcal{A}}$ е дефинирана така*

$$\Sigma_{\mathcal{A}} = \{(a, \lambda(q, a)) \mid (q, a) \in \mathrm{Dom}(\delta)\}$$

и $\mathcal{A} = (\Sigma_{\mathcal{A}}, Q, s, F, \delta_{\mathcal{A}})$ е тримованият краен детерминиран автомат, за който

$$\delta_{\mathcal{A}}(q, (a, b)) = \delta(q, a) \text{ за } q \in Q \text{ и } (a, b) \in \Sigma_{\mathcal{A}}.$$

Нека $\mathcal{A}' = (\Sigma_{\mathcal{A}}, Q', s', F', \delta'_{\mathcal{A}})$ е минимален автомат за $\mathcal{L}(\mathcal{A})$. Тогава преобразувателят $\mathcal{T}' = (\Sigma, \mathcal{R}_{\geq 0}, Q', s', F', \delta', \lambda', \iota)$, където

- $\delta' = \{(p, a, q) \mid (p, (a, b), q) \in \delta'_{\mathcal{A}}\}$;
- $\lambda' = \{(p, a, b) \mid (p, (a, b), q) \in \delta'_{\mathcal{A}}\}$,

е минимален преобразувател за $O_{\mathcal{T}}$. $\square$

Така получаваме, че всеки каноничен тримован преобразувател $\mathcal{T} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ може да бъде минимизиран за време $O(|\Sigma_{\mathcal{A}}| + |\mathrm{Dom}(\delta)| \log_2 |Q|)$ – времето, за което може да бъде минимизиран тримованият краен детерминиран автомат $\mathcal{A}$, индуциран от $\mathcal{T}$, и памет $O(|\Sigma_{\mathcal{A}}| + O(|\mathrm{Dom}(\delta)|))$.

6.3 Псевдоминимизация на канонични f-преобразуватели

За всеки минимален автомат $\mathcal{A}$ за $\mathcal{L}(\mathcal{A})$ е в сила, че $\mathcal{A}$ е краен детерминиран автомат с минимален размер (брой състояния плюс брой преходи) измежду всички крайни детерминирани автомати, еквивалентни на $\mathcal{A}$. Също така знаем, че задачата за минимизация на краен детерминиран автомат може да се реши ефективно.

В случая на минимален преобразувател са верни аналогични твърдения. Нещата не стоят така, когато става въпрос за f-преобразуватели. Задачата за минимизация на f-преобразувател (т.е. задачата за построяване на f-преобразувател с минимален размер измежду всички f-преобразуватели, еквивалентни на изходния) е NP-пълна (следва от резултата, представен в Björklund, Björklund и Zechner 2014). Поради тази причина ще

дефинираме понятието „псевдоминимален f-преобразувател" – f-преобразувател с минимален брой състояния измежду всички f-преобразуватели, еквивалентни на изходния, и ще покажем как можем да псевдоминимизираме определен вид f-преобразуватели.

Псевдоминимизацията на един f-преобразувател изисква първо той да бъде канонизиран. В предходната глава показахме как могат да бъдат канонизирани f-преобразуватели, чиито подлежащи преобразуватели имат изходи от $\mathcal{R}_{[0,1]}$. Също така показахме, че f-преобразувателите на ниво символи могат да бъдат канонизирани, въпреки че подлежащите им преобразуватели могат да имат изходи по-големи от 1. Аналогично на забележката от предишния раздел, тук също е в сила по-общо твърдение – всеки f-преобразувател $\mathcal{F}$ с изходи от $\mathcal{R}_{\geq 0}$, такъв че подлежащият му преобразувател $\mathcal{T}$ има изходи от $\mathcal{R}_{\geq 0}$ и $\mathrm{mso}_{\mathcal{T}}(q)$ е дефинирано за всяко състояние q на $\mathcal{T}$, може да бъде канонизиран. Поради тази причина, в тази глава под каноничен f-преобразувател ще разбираме f-преобразувател с изходи от $\mathcal{R}_{\geq 0}$, чийто подлежащ преобразувател е каноничен (тоест изходите му са от $\mathcal{R}_{\geq 0}$ и максималните изходи на състоянията му са дефинирани и равни на 1).

Определение 6.15. Нека $\mathcal{F}$ е f-преобразувател и $\mathcal{T}$ е подлежащият преобразувател на $\mathcal{F}$. Казваме, че $\mathcal{F}$ е *псевдоминимален* f-преобразувател за $O_{\mathcal{F}}$, ако $\mathcal{T}$ е минимален преобразувател за $O_{\mathcal{T}}$.

От дефиницията за минимален f-преобразувател $\mathcal{F}$ следва, че щом подлежащият му преобразувател $\mathcal{T}$ е преобразувателят с минимален размер, представящ функцията $O_{\mathcal{T}} = O_{\mathcal{F}}$, то $\mathcal{F}$ е f-преобразувателят с най-малък брой състояния, представящ $O_{\mathcal{F}}$. Ако допуснем противното, тоест, че съществува f-преобразувател $\mathcal{F}'$, еквивалентен на $\mathcal{F}$ и имащ по-малко състояния, то неговият подлежащ преобразувател $\mathcal{T}'$ би представял функцията $O_{\mathcal{T}'} = O_{\mathcal{F}'} = O_{\mathcal{F}} = O_{\mathcal{T}}$ и би имал по-малък размер от $\mathcal{T}$, което противоречи с минималността на $\mathcal{T}$.

Тримован f-преобразувател ще наричаме всеки f-преобразувател, чийто подлежащ преобразувател е тримован. От цитираните в предходния раздел резултати за минимизация на канонични преобразуватели следва, че можем да псевдоминимизираме каноничен f-преобразувател като първо го тримоваме и после приложим следното твърдение.

Твърдение 6.12. *Нека $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е каноничен тримован f-преобразувател и $\mathcal{T}$ е подлежащият преобразувател на $\mathcal{F}$. Нека $\mathcal{T}' = (\Sigma, \mathcal{R}_{\geq 0}, Q', s', F', \delta', \lambda', \iota)$ е минималният преобразувател, еквивалентен на $\mathcal{T}$, от твърдение 6.11 и $\xi\colon Q' \to Q$ е функция, която избира по един представител за всеки клас на еквивалентност в Q'. Тогава f-преобразувателят $\mathcal{F}' = (\Sigma, \mathcal{R}_{\geq 0}, Q', s', F', \delta'', f'', \lambda'', \varphi'', \iota)$, където*

- $\delta'' = \{(q, a, \delta'(q, a)) \mid q \in Q', (\xi(q), a) \in \mathrm{Dom}(\delta)\}$;
- $\lambda'' = \{(q, a, \lambda'(q, a)) \mid (q, a) \in \mathrm{Dom}(\delta'')\}$;
- $f'' = \{(q, [f(\xi(q))]_{\equiv_{\mathcal{T}}}) \mid q \in Q', \xi(q) \in \mathrm{Dom}(f)\}$;
- $\varphi'' = \{(q, \varphi(\xi(q))) \mid q \in \mathrm{Dom}(f'')\}$,

е псевдоминимален f-преобразувател за $O_{\mathcal{F}}$. □

Така получаваме, че всеки каноничен тримован f-преобразувател $\mathcal{F} = (\Sigma, \mathcal{R}_{\geq 0}, Q, s, F, \delta, \lambda, \iota)$ може да бъде псевдоминимизиран за време $O(|\Sigma_{\mathcal{A}}| + |Q||\Sigma| \log_2 |Q|)$ и памет $O(|\Sigma_{\mathcal{A}}| + |Q||\Sigma|)$.

6.4 Модификация на алгоритъма на Béal и Crochemore

Пространствената сложност на псевдоминимизацията на каноничен тримован f-преобразувател, представящ езиков модел, може да бъде подобрена лесно от $O(|\Sigma_{\mathcal{A}}| + |Q||\Sigma|)$ на $O(|\Sigma_{\mathcal{A}}| + |G(\mathcal{F})|)$. Идеята е да не представяме явно δ_f и δ_f^{-1} (съответно функцията на преходите на подлежащия преобразувател и нейната обратна релация), а да ги имаме в неявен вид под формата на графа $G(\mathcal{F})$ и неговия обратен $G(\mathcal{F})^{-1}$.

Сега ще покажем как можем да изразим δ_f^{-1} посредством δ^{-1} и f^{-1}. За целта ще използваме едно представяне на функцията δ_f описано в Mihov и Schulz, 2018 (in press).

Определение 6.16. Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. За $\sigma \in \Sigma$ дефинираме фунцкията $f^\star_\sigma \colon Q \to Q$, като най-малката относно включване функция $f'^\star_\sigma \colon Q \to Q$ дефинирана за $q \in Q$ така

$$f'^\star_\sigma(q) = \begin{cases} q, & \text{ако } !\,\delta(q, \sigma) \\ f'^\star_\sigma(f(q)), & \text{иначе} \end{cases}$$

Твърдение 6.13. *Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Тогава за всяко $q \in Q$ и $\sigma \in \Sigma$ е в сила, че*

$$\delta_f(q, \sigma) = \delta(f^\star_\sigma(q), \sigma). \qquad \square$$

Вече можем да изразим релацията δ_f^{-1} посредством релациите δ^{-1} и $f^{\star -1}_\sigma$.

Твърдение 6.14. *Нека $(\Sigma, \mathcal{M}, Q, s, F, \delta, f, \lambda, \varphi, \iota)$ е f-преобразувател. Тогава за всяко $q \in Q$ е в сила, че*

$$\delta_f^{-1}(q) = \bigcup_{(p,a) \in \delta^{-1}(q)} f_a^{\star -1}(p). \qquad \square$$

Модификацията, която следва да опишем в псевдокод, пресмята $f_a^{\star -1}(p)$, използвайки единствено $G(\mathcal{F})$ и $G(\mathcal{F})^{-1}$, чиито ребра са разделени според това дали съответстват на δ-преходи или f-преходи. Ребрата на $G(\mathcal{F})$ са представени посредством списъците на съседство Δ и Φ. Също така ще считаме, че имаме функция IsReal, която казва за всеки връх на $G(\mathcal{F})$ дали съответства на състояние от изходния f-преобразувател $\mathcal{F}$, или не.

Сега ще опишем промените, които трябва да бъдат направени в псевдокода, описващ алгоритъма на Béal и Crochemore.

---

function BéalCrochemoreMinimization((s, F, Δ, Φ))
  $(\mathcal{B}, \mathcal{I}) \leftarrow$ StateEquivalence(F, Δ, Φ)

От това, което казахме, следва, че BéalCrochemoreMinimization и StateEquivalence ще приемат двата аргумента Δ и Φ, които описахме по-горе. Промяната, която трябва да направим във функцията StateEquivalence, е освен Δ^{-1} да пресметнем и Φ^{-1}.

Съществената модификация е в процедурата RefinePartition (която вече приема като аргументи Δ^{-1} и Φ^{-1}), където трябва да намерим всички състояния, които влизат с дадена буква в състояние от разглеждания свидетел $\mathcal{B}[t]$. Тази модификация ще отделим във функцията DeltaFTransitions.

Функцията DeltaFTransitions намира по дадено състояние q множеството от всички двойки от буква и състояние (a, p), за които $\delta_f(p, a) = \delta(f^\star_a(p), a) = q$. Тъй като похватите, използвани в псевдокода на функцията DeltaFTransitions, са същите като тези от оригиналния алгоритъм на Béal и Crochemore, можем лесно да съобразим, че времевата

```
function StateEquivalence(F, Δ, Φ)
    ...
    Δ^{-1} ← ReverseGraph(Δ)
    Φ^{-1} ← ReverseGraph(Φ)
    ...
    RefinePartition((B, I, P, T, Y), (Σ', Σ'', σ, τ), δ^{-1}, Φ^{-1}, B[t])
    ...
```

```
procedure RefinePartition((B, I, P, T, Y), (Σ', Σ'', σ, τ), Δ^{-1}, Φ^{-1}, B)
    ...
    for all q ∈ B do
        for all (a, p) ∈ DeltaFTransitions(q, Δ^{-1}, Φ^{-1}, Σ'') do
    ...
```

сложност на функцията е $O(|T|)$, тоест е линейна относно размера на резултата $|\delta_f^{-1}(q)|$ и не нарушава времевата сложност на алгоритъма.

Идеята на имплементацията на DeltaFTransitions е следната. Списъкът T се създава на ред 2 и той ще съдържа резултата от извикването на функцията. **for** цикълът на ред 4 намира състоянията p, такива че $\delta(p, \sigma) = q$ за някое σ. Тези σ се пазят в списъка Σ''_i, а в $\Sigma''[\sigma]$ за всяко такова σ се пазят състоянията p, за които $\delta(p, \sigma) = q$. Всяка итерация на цикъла от ред 8 намира и прибавя към T всички двойки (a, p), такива че $p \in f_a^\star(q)$. Това става по следния начин. Първо, цикълът на ред 9 добавя към T всички (a, p), такива че $\delta(p, a) = q$. След това **while** цикълът разглежда всички състояния p, които могат да достигнат посредством f-преходи до състояние от $\Sigma''[a]$ и в същото време нямат дефиниран δ-преход с a. Разглеждането на състояния, за които IsReal връща лъжа, не влошава времевата сложност на функцията, тъй като на всяко такова състояние съответства друго състояние, за което IsReal връща истина и което бива добавено към резултата.

```
1: function DeltaFTransitions(q, Δ^{-1}, Φ^{-1}, Σ'')
2:     T ← NewList()
3:     Σ''_i ← NewList()
4:     for all (a, p) ∈ Δ^{-1} do
5:         if |Σ''[a]| = 0 then
6:             ListAppend(Σ''_i, a)
7:         ListAppend(Σ''[a], p)
8:     for all a ∈ Σ''_i do
9:         for all p ∈ Σ''[a] do
10:            if IsReal(p) then
11:                ListAppend(T, (a, p))
12:        Q ← Σ''[a]
13:        while |Q| ≠ 0 do
14:            p ← ListHead(Q)
15:            ListRemove(Q, p)
16:            for all r ∈ Φ^{-1}[p] do
17:                if IsReal(r) then
18:                    ListAppend(T, (a, r))
19:                ListAppend(Q, r)
20:        ListEmpty(Σ''[a])
21:    ListEmpty(Σ''_i)
22:    return T
```

Приложение А

В това приложение ще въведем понятията почти стохастична матрица и автоматна матрица и ще докажем някои техни свойства. Крайната цел е да се възползваме от тези понятия и техните свойства в приложение Б.

Множеството от всички квадратни матрици от ред n над полето F ще бележим с $M_n(F)$. Ще използваме главни латински букви (като A), за да означаваме матрици. Елементът на матрицата $A \in M_n(F)$ от ред $1 \leq i \leq n$ и колона $1 \leq j \leq n$ ще бележим със същата латинска буква като матрицата, която обаче е малка и има индекси i и j (в случая a_{ij}). Матриците, които ние ще разглеждаме, ще бъдат основно над полето на реалните числа $\mathbb{R}$.

Определение А.1. Нека $A \in M_n(\mathbb{R})$. *Път* в A наричаме всяка редица $\pi = (p_1, p_2, \ldots, p_{m+1})$ от елементи на $\{1, 2, \ldots, n\}$, такава че

$$(\forall 1 \leq i \leq m)(a_{p_i p_{i+1}} \neq 0).$$

p_1 наричаме *начало*, p_{m+1} – *край*, а числото m – *дължина* на пътя π.

Определение А.2. Нека $A \in M_n(\mathbb{R})$, $1 \leq i, j \leq n$ и $k \in \mathbb{N}$. С $\Pi_A(i, j, k)$ ще отбелязваме множеството от всички пътища в A с начало i, край j и дължина k.

За удобство ще въведем следните означения, свързани с пътища в матрица $A \in M_n(\mathbb{R})$,

- $\Pi_A(i, j) = \bigcup_{k=0}^{\infty} \Pi_A(i, j, k)$ за $1 \leq i, j \leq n$;
- $\Pi_A(i, J) = \bigcup_{j \in J} \Pi_A(i, j)$ за $1 \leq i \leq n$ и $J \subseteq \{1, 2, \ldots, n\}$;
- $\Pi_A(i) = \Pi_A(i, \{1, 2, \ldots, n\})$ за $1 \leq i \leq n$;
- $\Pi_A = \bigcup_{i=1}^{n} \Pi_A(i)$.

Ще си позволяваме да изпускаме долния индекс на Π във въведените горе означения, когато това не води до недоразумения (например, когато говорим само за една матрица).

Определение А.3. Нека $A \in M_n(\mathbb{R})$. Фукцията $w \colon \Pi \to \mathbb{R}$ наричаме *теглова функция* и дефинираме за път $\pi = (p_1, p_2, \ldots, p_{m+1})$ в A така

$$w(\pi) = \prod_{i=1}^{m} a_{p_i p_{i+1}}.$$

$w(\pi)$ наричаме *тегло* на пътя π.

Твърдение А.1. *Нека $A \in M_n(\mathbb{R})$ и $k \in \mathbb{N}$. Тогава за матрицата A^k е в сила, че*

$$(\forall 1 \leq i \leq n)(\forall 1 \leq j \leq n)(a_{ij}^{(k)} = \sum_{\pi \in \Pi(i,j,k)} w(\pi)),$$

където с $a_{ij}^{(k)}$ за $1 \leq i, j \leq n$ бележим елементите на A^k.

Доказателство. Твърдението ще докажем с индукция по k.

$k = 0$

$$A^0 = I_n,$$

където с I_n означаваме единичната матрица от ред n. Тогава за $1 \leq i, j \leq n$

$$a_{ij}^{(0)} = \begin{cases} 1, & \text{ако } i = j \\ 0, & \text{иначе} \end{cases} = \begin{cases} \sum_{\pi \in \Pi(i,j,0)} 1, & \text{ако } i = j \\ \sum_{\pi \in \Pi(i,j,0)} 0, & \text{иначе} \end{cases} = \begin{cases} \sum_{\pi \in \Pi(i,j,0)} w(\pi), & \text{ако } i = j \\ \sum_{\pi \in \Pi(i,j,0)} w(\pi), & \text{иначе} \end{cases}$$

Горните равенства за верни, тъй като $\Pi(i,i,0) = \{(i)\}$ и $w((i)) = 1$, а за $i \neq j$ е в сила, че $\Pi(i,j,0) = \varnothing$.

$k = 1$

За $1 \leq i, j \leq n$ съществува път с начало i, край j и дължина 1 тогава и само тогава, когато $a_{ij} \neq 0$. Нещо повече, ако съществува такъв път, той е единствен и има тегло a_{ij}. Сега, тъй като

$$A^1 = A,$$

е в сила, че за $1 \leq i, j \leq n$

$$a_{ij}^{(1)} = a_{ij} = \sum_{\pi \in \Pi(i,j,1)} a_{ij} = \sum_{\pi \in \Pi(i,j,1)} w(\pi).$$

$k \rightsquigarrow k + 1$

Нека твърдението е в сила за $k \in \mathbb{N}$. Тогава

$$A^{k+1} = A^k A$$

и за $1 \leq i, j, \leq n$ е в сила, че

$$\begin{aligned} a_{ij}^{(k+1)} &= \sum_{t=1}^{n} a_{it}^{(k)} a_{tj} = \sum_{t=1}^{n} \left[\sum_{\pi \in \Pi(i,t,k)} w(\pi) \right] a_{tj} = \sum_{t=1}^{n} \sum_{\pi \in \Pi(i,t,k)} w(\pi) a_{tj} \\ &= \sum_{\substack{1 \leq t \leq n: \\ a_{tj} > 0}} \sum_{\pi \in \Pi(i,t,k)} w(\pi) a_{tj} = \sum_{\pi \in \Pi(i,j,k+1)} w(\pi). \end{aligned}$$

Последното равенство е вярно, тъй като

$$\pi \in \Pi(i,j,k+1) \iff (\exists 1 \leq t \leq n)(\exists \pi' \in \Pi(i,t,k))(a_{tj} \neq 0 \wedge \pi = \pi' j)$$

и теглото на пътя π е точно произведението на теглото на π' и a_{tj}. □

Определение А.4. Матрица $A \in M_n(\mathbb{R})$ наричаме *почти стохастична*, ако

- $(\forall 1 \leq i \leq n)(\forall 1 \leq j \leq n)(a_{ij} \geq 0)$;

- $(\forall 1 \leq i \leq n)(\sum_{j=1}^{n} a_{ij} \in \{0, 1\})$.

Определение А.5. Матрица $A \in M_n(\mathbb{R})$ наричаме *автоматна*, ако съществуват $1 \leq s \leq n$ и $\varnothing \subset F \subseteq \{1, 2, \ldots, n\}$, такива че

- $(\forall 1 \leq i \leq n)(\Pi(s, i) \neq \varnothing)$;
- $(\forall 1 \leq i \leq n)(\Pi(i, F) \neq \varnothing)$;
- $(\forall i \in F)(\sum_{j=1}^{n} a_{ij} = 0)$.

s наричаме *начало* на A, а елементите на F – *краища* на A.

Директно от дефинициите на почти стохастична и автоматна матрица се вижда следното свойство.

Твърдение А.2. *Нека $A \in M_n(\mathbb{R})$ е почти стохастична автоматна матрица. Тогава е в сила, че*

$$(\forall 1 \leq i \leq n)(\sum_{j=1}^{n} a_{ij} = 0 \iff i \in F). \qquad \square$$

Вектор-редовете ще означаваме с малки латински букви със стрелка отгоре (като $\vec{v}$). Вектор-стълбовете ще означаваме, както вектор-редовете, но с добавен горен индекс T (като $\vec{v}^T$). i-тия елемент на вектор-реда $\vec{v}$ (или вектор-стълба $\vec{v}^T$) ще бележим с $\vec{v}_i$ (или $\vec{v}_i^T$).

Твърдение А.3. *Нека $A \in M_n(\mathbb{R})$ е почти стохастична автоматна матрица. Тогава всички собствени стойности на A са по модул по-малки от 1.*

Доказателство. Да допуснем, че $\lambda \in \mathbb{C}$ е собствена стойност на A и $|\lambda| \geq 1$. Нека $\vec{v}$ е собствен вектор на A, съответстващ на λ. Тогава

$$A\vec{v}^T = \lambda\vec{v}$$

$$\left|\begin{array}{l} a_{11}\vec{v}_1 + a_{12}\vec{v}_2 + \ldots + a_{1n}\vec{v}_n = \lambda\vec{v}_1 \\ a_{21}\vec{v}_1 + a_{22}\vec{v}_2 + \ldots + a_{2n}\vec{v}_n = \lambda\vec{v}_2 \\ \ldots \\ a_{n1}\vec{v}_1 + a_{n2}\vec{v}_2 + \ldots + a_{nn}\vec{v}_n = \lambda\vec{v}_n \end{array}\right.$$

Нека $1 \leq i \leq n$ е такова, че $|\vec{v}_i| = \max\{|\vec{v}_j| \mid 1 \leq j \leq n\}$. С индукция по m ще докажем, че

$$(\forall m \in \mathbb{N})(\forall 1 \leq j \leq n)(\Pi(i, j, m) \neq \varnothing \implies |\vec{v}_j| = |\vec{v}_i|).$$

$m = 0$
Нека $1 \leq j \leq n$ е такова, че $\Pi(i, j, 0) \neq \varnothing$. Тогава $j = i$ и е в сила, че $|\vec{v}_j| = |\vec{v}_i|$.

$m \rightsquigarrow m + 1$
Нека $1 \leq j \leq n$ е такова, че $\Pi(i, j, m + 1) \neq \varnothing$ и твърдението е в сила за всяко $1 \leq k \leq n$, такова че $\Pi(i, k, m) \neq \varnothing$. Нека $\pi \in \Pi(i, j, m + 1)$, тоест $\pi = (i, p_1, p_2, \ldots, p_m, j)$. Тогава пътят $(i, p_1, p_2, \ldots, p_m) \in \Pi(i, p_m, m)$ и от индукционното предположение следва, че $|\vec{v}_{p_m}| = |\vec{v}_i|$. Също така пътят π свидетелства за това, че $a_{p_m j} > 0$.

Ще докажем, че $|\vec{v}_j| = |\vec{v}_i|$. За целта да допуснем, че $|\vec{v}_j| < |\vec{v}_i|$. Тогава

$$\begin{aligned}
|\lambda||\vec{v}_i| = |\lambda||\vec{v}_{p_m}| = |\lambda\vec{v}_{p_m}| &= |a_{p_m1}\vec{v}_1 + a_{p_m2}\vec{v}_2 + \ldots + a_{p_mn}\vec{v}_n| \\
&\leq |a_{p_m1}\vec{v}_1| + |a_{p_m2}\vec{v}_2| + \ldots + |a_{p_mn}\vec{v}_n| \\
&= a_{p_m1}|\vec{v}_1| + a_{p_m2}|\vec{v}_2| + \ldots + a_{p_mn}|\vec{v}_n| \\
&< a_{p_m1}|\vec{v}_i| + a_{p_m2}|\vec{v}_i| + \ldots + a_{p_mn}|\vec{v}_i| \\
&= \begin{cases} |\vec{v}_i|, & \text{ако } \sum_{j=1}^{n} a_{p_mj} = 1 \\ 0, & \text{ако } \sum_{j=1}^{n} a_{p_mj} = 0 \end{cases}
\end{aligned}$$

1сл. $\boldsymbol{\sum_{j=1}^{n} a_{p_mj} = 0}$
Този случай е невъзможен, тъй като тогава бихме имали, че $|\lambda||\vec{v}_i| < 0$.

2сл. $\boldsymbol{\sum_{j=1}^{n} a_{p_mj} = 1}$
Тогава $|\lambda||\vec{v}_i| < |\vec{v}_i|$, което противоречи с допускането, че $|\lambda| \geq 1$. Следователно едно от двете ни допускания е неправилно. Ако първото допускане е грешно, то доказваме, че $|\lambda| < 1$. Затова нека второто допускане е грешно, тоест $|\vec{v}_j| = |\vec{v}_i|$.

Тъй като A е автоматна матрица, то $\Pi(i, F) \neq \varnothing$. Следователно съществува $f \in F$, такова че $|\vec{v}_f| = |\vec{v}_i|$. Но $\sum_{j=1}^{n} a_{fj} = 0$, което значи, че $|\vec{v}_f| = 0$. Така получаваме, че $(\forall 1 \leq j \leq n)(|\vec{v}_j| = 0)$, тоест $\vec{v} = \vec{o}$, което противоречи с избора на $\vec{v}$. Следователно допускането ни е грешно и $|\lambda| < 1$. □

Твърдение А.4. *Нека $A \in M_n(\mathbb{R})$ е почти стохастична автоматна матрица. Тогава матрицата $I_n - A$ е обратима и*

$$(I_n - A)^{-1} = \sum_{i=0}^{\infty} A^i.$$

Доказателство. A е матрица над полето $\mathbb{R}$. Следователно всички характеристични корени на A са комплексни числа. Ако разгледаме A като матрица над полето $\mathbb{C}$, всичките ѝ характеристични корени ще бъдат в нейното поле. Тогава знаем от линейната алгебра, че A е подобна на жорданова матрица. Тоест

$$A = T^{-1}DT,$$

където $T \in M_n(\mathbb{C})$ е матрица на смяна на базиса, а $D \in M_n(\mathbb{C})$ е жорданова матрица. Тоест

$$D = \begin{pmatrix} D_{n_1}(\lambda_1) & & & 0 \\ & D_{n_2}(\lambda_2) & & \\ & & \ddots & \\ 0 & & & D_{n_k}(\lambda_k) \end{pmatrix},$$

където $D_{n_i}(\lambda_i)$ е жорданова клетка от ред n_i, съответстваща на собствената стойност λ_i на матрицата A, за $1 \leq i \leq k$ и $\sum_{i=1}^{k} n_i = n$. Тогава

$$\sum_{i=0}^{\infty} A^i = \sum_{i=0}^{\infty} (T^{-1}DT)^i = \sum_{i=0}^{\infty} T^{-1}D^iT = T^{-1}\left(\sum_{i=0}^{\infty} D^i\right)T.$$

Тъй като D е жорданова матрица, за $i \in \mathbb{N}$

$$D^i = \begin{pmatrix} D_{n_1}(\lambda_1) & & & 0 \\ & D_{n_2}(\lambda_2) & & \\ & & \ddots & \\ 0 & & & D_{n_k}(\lambda_k) \end{pmatrix}^i = \begin{pmatrix} D_{n_1}(\lambda_1)^i & & & 0 \\ & D_{n_2}(\lambda_2)^i & & \\ & & \ddots & \\ 0 & & & D_{n_k}(\lambda_k)^i \end{pmatrix}.$$

С индукция по i ще докажем, че за всяко $1 \leq j \leq k$

$$D_{n_j}(\lambda_j)^i = \begin{pmatrix} \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} & \binom{i}{2}\lambda_j^{i-2} & \cdots & \binom{i}{n_j-1}\lambda_j^{i-n_j+1} \\ & \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} & \cdots & \binom{i}{n_j-2}\lambda_j^{i-n_j+2} \\ & & \ddots & & \vdots \\ & & & \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} \\ 0 & & & & \lambda_j^i \end{pmatrix}.$$

$i = 0$

$$D_{n_j}(\lambda_j)^0 = I_{n_j} = \begin{pmatrix} \lambda_j^0 & \binom{0}{1}\lambda_j^{0-1} & \binom{0}{2}\lambda_j^{0-2} & \cdots & \binom{0}{n_j-1}\lambda_j^{0-n_j+1} \\ & \lambda_j^0 & \binom{0}{1}\lambda_j^{i-1} & \cdots & \binom{0}{n_j-2}\lambda_j^{0-n_j+2} \\ & & \ddots & & \vdots \\ & & & \lambda_j^0 & \binom{0}{1}\lambda_j^{0-1} \\ 0 & & & & \lambda_j^0 \end{pmatrix}.$$

$i \rightsquigarrow i+1$

Нека твърдението е в сила за $i \in \mathbb{N}$. Тогава

$$D_{n_j}(\lambda_j)^{i+1} = D_{n_j}(\lambda_j)^i D_{n_j}(\lambda_j) =$$
$$\begin{pmatrix} \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} & \binom{i}{2}\lambda_j^{i-2} & \cdots & \binom{i}{n_j-1}\lambda_j^{i-n_j+1} \\ & \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} & \cdots & \binom{i}{n_j-2}\lambda_j^{i-n_j+2} \\ & & \ddots & & \vdots \\ & & & \lambda_j^i & \binom{i}{1}\lambda_j^{i-1} \\ 0 & & & & \lambda_j^i \end{pmatrix} \begin{pmatrix} \lambda_j & 1 & & 0 \\ & \lambda_j & 1 & \\ & & \ddots & 1 \\ 0 & & & \lambda_j \end{pmatrix}.$$

Ясно е, че елементите на $D_{n_j}(\lambda_j)^{i+1}$ от главния диагонал ще бъдат равни на λ_j^{i+1}, а тези под него на 0. Нека $1 \leq s < n_j$ и $s < t \leq n_j$. Искаме да покажем, че елементът на $D_{n_j}(\lambda_j)^{i+1}$ от ред s и колона t е равен на

$$\binom{i+1}{t-s}\lambda_j^{i-(t-s)+1}.$$

s-тият вектор-ред на $D_{n_j}(\lambda_j)^{i+1}$ е равен на

$$\begin{pmatrix} 0 & \cdots & 0 & \binom{i}{0}\lambda_j^{i-0} & \binom{i}{1}\lambda_j^{i-1} & \cdots & \binom{i}{n_j-t}\lambda_j^{i-n_j+t} \end{pmatrix}$$

и има $s-1$ нули в началото. t-тият вектор-стълб на $D_{n_j}(\lambda_j)$ е равен на

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \lambda_j \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

и има $t-2$ нули в началото. Тогава тяхното скаларно произведение е равно на

$$\begin{aligned}\binom{i}{t-s-1}\lambda_j^{i-(t-s)+1} + \binom{i}{t-s}\lambda_j^{i-(t-s)}\lambda_j &= \lambda_j^{i-(t-s)+1}\left(\binom{i}{t-s-1} + \binom{i}{t-s}\right) \\ &= \binom{i+1}{t-s}\lambda_j^{i-(t-s)+1},\end{aligned}$$

с което завършваме индукционната стъпка.

Сега имаме, че

$$\sum_{i=0}^{\infty} D^i = \begin{pmatrix} \sum_{i=0}^{\infty} D_{n_1}(\lambda_1)^i & & & 0 \\ & \sum_{i=0}^{\infty} D_{n_2}(\lambda_2)^i & & \\ & & \ddots & \\ 0 & & & \sum_{i=0}^{\infty} D_{n_k}(\lambda_k)^i \end{pmatrix},$$

а за всяко $1 \leq j \leq k$

$$\sum_{i=0}^{\infty} D_{n_j}(\lambda_j)^i = \begin{pmatrix} \frac{1}{1-\lambda_j} & \frac{1}{(1-\lambda_j)^2} & \frac{1}{(1-\lambda_j)^3} & \cdots & \frac{1}{(1-\lambda_j)^{n_j-1}}\frac{1}{(1-\lambda_j)^{n_j}} \\ 0 & \frac{1}{1-\lambda_j} & \frac{1}{(1-\lambda_j)^2} & \cdots & \frac{1}{(1-\lambda_j)^{n_j-1}} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{1-\lambda_j} & \frac{1}{(1-\lambda_j)^2} \\ 0 & 0 & \cdots & 0 & \frac{1}{1-\lambda_j} \end{pmatrix}.$$

Последното е вярно, тъй като

- за всяко $t \in \mathbb{N}$ формалният ред

$$\sum_{i=0}^{\infty} \binom{i}{t}\lambda_j^{i-t}$$

е сходящ, поради това че

$$\lim_{i\to\infty}\left|\frac{\binom{i+1}{t}\lambda_j^{i+1-t}}{\binom{i}{t}\lambda_j^{i-t}}\right| = |\lambda_j| < 1;$$

- формалният ред се представя като

$$\sum_{i=0}^{\infty}\binom{i}{t}\lambda_j^{i-t} = t!\left(\sum_{i=0}^{\infty}\lambda_j^i\right)^{(t)} = t!\left(\frac{1}{1-\lambda_j}\right)^{(t)} = \frac{1}{(1-\lambda_j)^t},$$

където $f(\lambda_j)^{(t)}$ е t-тата производна на f относно λ_j.

Сега директно се проверява, че за всяко $1 \leq j \leq k$

$$\sum_{i=0}^{\infty} D_{n_j}(\lambda_j)^i = (I_{n_j} - D_{n_j}(\lambda_j))^{-1},$$

от което следва, че

$$\sum_{i=0}^{\infty} D^i = (I_n - D)^{-1}.$$

Връщайки се обратно в началото, получаваме

$$\begin{aligned}
\sum_{i=0}^{\infty} A^i &= T^{-1}\left(\sum_{i=0}^{\infty} D^i\right) T \\
&= T^{-1}(I_n - D)^{-1}T \\
&= ((I_n - D)T)^{-1}T \\
&= ((I_n - D)T)^{-1}(T^{-1})^{-1} \\
&= (T^{-1}(I_n - D)T)^{-1} \\
&= (I_n - T^{-1}DT)^{-1} \\
&= (I_n - A)^{-1}.
\end{aligned}$$

□

Твърдение А.5. *Нека $A \in M_n(\mathbb{R})$ е почти стохастична автоматна матрица с начало s и краища F. Тогава за всяко $1 \leq q \leq n$ е в сила, че*

$$\vec{q}\left(\sum_{k=0}^{\infty} A^k\right)\vec{f}^T = \sum_{\pi \in \Pi(q,F)} w(\pi),$$

където $\vec{q}$ е характеристичният вектор на $\{q\}$, а $\vec{f}$ е характеристичният вектор на F.

Доказателство. Нека с A^∞ да означим матрицата $\sum_{k=0}^{\infty} A^k$. От твърдение А.1 следва, че за всеки елемент на A^∞

$$a_{ij}^\infty = \sum_{\pi \in \Pi(i,j)} w(\pi).$$

Тогава

$$\vec{q}\left(\sum_{k=0}^{\infty} A^k\right)\vec{f}^T = \begin{pmatrix} a_{q1}^\infty & a_{q2}^\infty & \dots & a_{qn}^\infty \end{pmatrix}\vec{f}^T = \sum_{j \in F} a_{qj}^\infty = \sum_{j \in F}\sum_{\pi \in \Pi(q,j)} w(\pi) = \sum_{\pi \in \Pi(q,F)} w(\pi).$$ □

Твърдение А.6. *Нека $A \in M_n(\mathbb{R})$ е почти стохастична автоматна матрица с начало s и краища F. Тогава за всяко $1 \leq q \leq n$ е в сила, че*

$$\sum_{\pi \in \Pi(q,F)} w(\pi) = 1.$$

Доказателство. Нека $\vec{q}$ е характеристичният вектор на $\{q\}$, а $\vec{f}$ е характеристичният вектор на F. Нека $\vec{q}(I_n - A)^{-1} = \vec{u}$, тоест $\vec{q} = \vec{u}(I_n - A)$. Тогава

$$\vec{q}_i = \vec{u}_i - \sum_{j=0}^{n} \vec{u}_j a_{ji}$$

$$\sum_{i=0}^{n} \vec{q}_i = \sum_{i=0}^{n} \left(\vec{u}_i - \sum_{j=0}^{n} \vec{u}_j a_{ji} \right)$$
$$1 = \sum_{i=0}^{n} \vec{u}_i - \sum_{j=0}^{n} \vec{u}_j \sum_{i=0}^{n} a_{ji} = \sum_{i=0}^{n} \vec{u}_i - \sum_{i=0}^{n} \vec{u}_i \sum_{j=0}^{n} a_{ij}$$
$$1 = \sum_{i=0}^{n} \vec{u}_i \left(1 - \sum_{j=0}^{n} a_{ij} \right).$$

Ние знаем, че

$$1 - \sum_{j=0}^{n} a_{ij} = \begin{cases} 1, & \text{ако } \vec{f}_i = 1, \text{ тоест } i \in F \\ 0, & \text{иначе} \end{cases}$$

Така получаваме, че $\vec{u}\vec{f}^T = 1$, от което, използвайки твърдение A.4 и твърдение A.5, следва, че

$$1 = \vec{u}\vec{f}^T = \vec{q}(I_n - A)^{-1}\vec{f}^T = \vec{q}\left(\sum_{k=0}^{\infty} A^k \right) \vec{f}^T = \sum_{\pi \in \Pi(q,F)} w(\pi). \qquad \square$$

Приложение Б

В това приложение ще докажем необходимото и достатъчно условие функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0,1]$ да определя n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, тоест да принадлежи на Ξ_P, формулирано в твърдение 1.3.

Първо ще разгледаме случая, в който $n = 1$.

Твърдение Б.1. *Нека $\widetilde{\Sigma}$ е азбука и ξ е функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{\varepsilon\}$ в $[0,1]$. Тогава съществува 1-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$ тогава и само тогава, когато*

- $\xi(\$, \varepsilon) > 0$;
- $\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \varepsilon) = 1$.

Доказателство.

$\Rightarrow$ Нека P е 1-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$. Тогава съществува дума $\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \dots a_{m+2}$, такава че $P(\{\alpha\}) > 0$. Вероятността на α можем да запишем така

$$P(\{\alpha\}) = \prod_{i=2}^{m+2} \xi(a_i, \varepsilon) = \left(\prod_{i=2}^{m+1} \xi(a_i, \varepsilon) \right) \xi(\$, \varepsilon) > 0,$$

което означава, че $\xi(\$, \varepsilon) > 0$. Също така условната вероятност $P(B_{\hat{}a} \mid B_{\hat{}})$ е дефинирана за всяко $a \in \widetilde{\Sigma} \cup \{\$\}$, от което следва, че

$$\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \varepsilon) = \sum_{a \in \widetilde{\Sigma} \cup \{\$\}} P(B_{\hat{}a} \mid B_{\hat{}}) = P\Big(\bigcup_{a \in \widetilde{\Sigma} \cup \{\$\}} B_{\hat{}a} \mid B_{\hat{}} \Big) = P(B_{\hat{}} \mid B_{\hat{}}) = 1.$$

$\Leftarrow$ Нека ξ е такава, че $\xi(\$, \varepsilon) > 0$ и $\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \varepsilon) = 1$. Да разгледаме изброимо адитивната функция $P \colon \mathcal{P}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \to [0,1]$, дефинирана за $\{\alpha\} \in \mathcal{P}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})$, $\alpha = a_1 a_2 \dots a_{m+2}$ така

$$P(\{\alpha\}) = \prod_{i=2}^{m+2} \xi(a_i, \varepsilon).$$

За да докажем, че P е езиков модел остава да покажем, че

$$\sum_{\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}} P(\{\alpha\}) = 1.$$

Използвайки това, че $\sum_{a\in\widetilde{\Sigma}\cup\{\$\}}\xi(a,\varepsilon)=1$, с индукция по $i\in\mathbb{N}^+$ лесно се вижда, че

$$\sum_{a_1,a_2,\ldots,a_i\in\widetilde{\Sigma}}\prod_{j=1}^{i}\xi(a_j\mid\varepsilon)=\sum_{a_1\in\widetilde{\Sigma}}\xi(a_1\mid\varepsilon)\sum_{a_2\in\widetilde{\Sigma}}\xi(a_2\mid\varepsilon)\cdots\sum_{a_i\in\widetilde{\Sigma}}\xi(a_i\mid\varepsilon)=(1-\xi(\$\mid\varepsilon))^i,$$

от което следва, че

$$\begin{aligned}
\sum_{\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}}P(\{\alpha\})&=\sum_{i=0}^{\infty}\sum_{\substack{a_1a_2\ldots a_{i+2}\in\\ \{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}}}P(\{a_1a_2\ldots a_i\})\\
&=\sum_{i=0}^{\infty}\sum_{\substack{a_1a_2\ldots a_{i+2}\in\\ \{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}}}\prod_{j=2}^{i+2}\xi(a_j,\varepsilon)\\
&=\sum_{i=0}^{\infty}\sum_{\substack{a_1a_2\ldots a_{i+2}\in\\ \{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}}}\left(\prod_{j=2}^{i+1}\xi(a_j\mid\varepsilon)\right)\xi(\$\mid\varepsilon)\\
&=\sum_{i=0}^{\infty}\xi(\$\mid\varepsilon)\sum_{\substack{a_1a_2\ldots a_{i+2}\in\\ \{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}}}\prod_{j=2}^{i+1}\xi(a_j\mid\varepsilon)\\
&=\xi(\$\mid\varepsilon)+\xi(\$\mid\varepsilon)\sum_{i=1}^{\infty}\sum_{a_1,a_2,\ldots,a_i\in\widetilde{\Sigma}}\prod_{j=1}^{i}\xi(a_j\mid\varepsilon)\\
&=\xi(\$\mid\varepsilon)+\xi(\$\mid\varepsilon)\sum_{i=1}^{\infty}(1-\xi(\$\mid\varepsilon))^i\\
&=\xi(\$\mid\varepsilon)\sum_{i=0}^{\infty}(1-\xi(\$\mid\varepsilon))^i\\
&=\xi(\$\mid\varepsilon)\frac{1}{1-(1-\xi(\$\mid\varepsilon))}\\
&=1.
\end{aligned}$$

Сега ще покажем, че за всяко $a\in\widetilde{\Sigma}\cup\{\$\}$ и всяко $\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star$, такова че $P(B_\alpha)>0$, е в сила, че

$$P(B_{\alpha a}\mid B_\alpha)=\xi(a,\varepsilon),$$

от което следва, че P е 1-грамен езиков модел и $\xi\in\Xi_P$. Наистина нека $a\in\widetilde{\Sigma}\cup\{\$\}$ и $\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star$, $\alpha=a_1a_2\ldots a_{m+1}$ е такова, че $P(B_\alpha)>0$. Тогава

$$P(B_{\alpha a}\mid B_\alpha)=\frac{P(B_{\alpha a})}{P(B_\alpha)}=\frac{\sum_{\beta\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}:\ \alpha a\preceq\beta}P(\{\beta\})}{\sum_{\beta\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}:\ \alpha\preceq\beta}P(\{\beta\})}.$$

Числителят е равен на

$$\begin{cases}\left(\prod_{i=2}^{m+1}\xi(a_i,\varepsilon)\right)\xi(a,\varepsilon), & \text{ако } a=\$\\ \left(\prod_{i=2}^{m+1}\xi(a_i,\varepsilon)\right)\xi(a,\varepsilon)\sum_{i=0}^{\infty}\sum_{b_1b_2\ldots b_{i+1}\in\widetilde{\Sigma}^\star\circ\{\$\}}\left(\prod_{j=1}^{i}\xi(b_j,\varepsilon)\right)\xi(\$,\varepsilon), & \text{иначе}\end{cases}$$

а знаменателят е равен на

$$\left(\prod_{i=2}^{m+1}\xi(a_i,\varepsilon)\right)\sum_{i=0}^{\infty}\sum_{\substack{b_1b_2\ldots b_{i+1}\in\\ \widetilde{\Sigma}^\star\circ\{\$\}}}\left(\prod_{j=1}^{i}\xi(b_j,\varepsilon)\right)\xi(\$,\varepsilon).$$

Сега, тъй като

$$\sum_{i=0}^{\infty}\sum_{\substack{b_1b_2\ldots b_{i+1}\in\\ \widetilde{\Sigma}^\star\circ\{\$\}}}\left(\prod_{j=1}^{i}\xi(b_j,\varepsilon)\right)\xi(\$,\varepsilon)=1,$$

което показахме по-горе, следва, че

$$P(B_{\alpha a}\mid B_\alpha)=\xi(a,\varepsilon).\qquad\square$$

За $n>1$ ще подходим по различен начин. Всъщност случаят $n=1$ също влиза в общия подход, който ще представим за $n>1$, но го показахме отделно, тъй като е по-прост и не изисква използването на твърденията от приложение А.

Определение Б.1. Нека $A\in M_n(\mathbb{R})$, $1\le s\le n$ и $F\subseteq\{1,2,\ldots,n\}$. Нека

$$Q=\{1\le i\le n\mid \Pi_A(s,i)\neq\varnothing\wedge\Pi_A(i,F)\neq\varnothing\}$$

и $\iota\colon Q\to\{1,2,\ldots,|Q|\}$ е биекция. Тогава с $\mathrm{trim}(A,s,F)$ означаваме множеството от матрици от $M_{|Q|}(\mathbb{R})$, такова че за всяка матрица B от $\mathrm{trim}(A,s,F)$

$$(\forall 1\le i\le|Q|)(\forall 1\le j\le|Q|)(b_{ij}=a_{\iota^{-1}(i)\iota^{-1}(j)}).$$

Директно от определение Б.1 се вижда, че следното твърдение е вярно.

Твърдение Б.2. *Нека $A\in M_n(\mathbb{R})$, $1\le s\le n$ и $F\subseteq\{1,2,\ldots,n\}$. Тогава за матрицата $B\in\mathrm{trim}(A,s,F)$ от ред m е в сила, че*

$$(\forall 1\le i\le m)(\Pi_B(\iota(s),i)\neq\varnothing\wedge\Pi_B(i,\{\iota(j)\mid j\in F\cap\mathrm{Dom}(\iota)\})\neq\varnothing),$$

където ι е биекцията от дефиницията на B. $\square$

Определение Б.2. Нека $\widetilde{\Sigma}$ е азбука и $n>1$. Нека ξ е функция от $(\widetilde{\Sigma}\cup\{\$\})\times\{h_n(\alpha)\mid\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\}$ в $[0,1]$ и ι е биекция между множествата $\{h_n(\alpha)\mid\alpha\in\mathrm{Pref}(\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\})\setminus\{\varepsilon\}\}$ и $\{1,2,\ldots,m\}$. *Матрица на ξ* наричаме матрица $A\in M_m(\mathbb{R})$, такава че за всеки $1\le i,j\le m$

$$a_{ij}=\begin{cases}\xi(a,\iota^{-1}(i)), & \text{ако } \$\not\equiv\iota^{-1}(i)\wedge(\exists b\in\widetilde{\Sigma}\cup\{\$\})(h_n(\iota^{-1}(i)b)=\iota^{-1}(j))\\ 0, & \text{иначе}\end{cases}$$

където a е единственият елемент на $\widetilde{\Sigma}\cup\{\$\}$, такъв че $h_n(\iota^{-1}(i)a)=\iota^{-1}(j)$.

Определение Б.3. Нека $\widetilde{\Sigma}$ е азбука и $n>1$. Нека ξ е функция от $(\widetilde{\Sigma}\cup\{\$\})\times\{h_n(\alpha)\mid\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\}$ в $[0,1]$. Нека също така

- A е матрицата на ξ;
- ι_1 е биекцията от дефиницията на A;

- $s = \iota_1(\hat{})$;
- $F = \{\iota_1(h_n(\alpha)) \mid \alpha \in \mathrm{Pref}(\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}) \wedge \$ \succeq \alpha\}$;

Тогава въвеждаме следните означения

- A_ξ е коя да е матрица от $\mathrm{trim}(A, s, F)$;
- $s_\xi = \iota_2(s)$;
- $F_\xi = \{\iota_2(i) \mid i \in F \cap \mathrm{Dom}(\iota_2)\}$;
- $\iota_\xi = \iota_1 \circ \iota_2$,

където ι_2 е биекцията от дефиницията на A_ξ. Ще считаме, че операцията „композиция на функции" е дефинирана така, че $\iota_1 \circ \iota_2$ е функция от $\mathrm{Dom}(\iota_1)$ в $\mathrm{Rng}(\iota_2)$.

Тъй като до края на това приложение няма да разглеждаме едновременно повече от една функция $\xi\colon (\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\} \to [0,1]$, ще пропускаме индексите в означенията A_ξ, s_ξ, F_ξ и ι_ξ, въведени в определение Б.3.

Преди да преминем към основното твърдение, ще докажем няколко помощни твърдения.

Твърдение Б.3. *Нека $\widetilde{\Sigma}$ е азбука и $n > 1$. Нека P е n-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ и $\xi \in \Xi_P$. Тогава матрицата $A \in M_m(\mathbb{R})$ е автоматна с начало s и краища F.*

Доказателство. Щом P е n-грамен езиков модел над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, то съществува $\alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \dots a_{k+2}$, такова че

$$P(\{\alpha\}) = \prod_{i=2}^{k+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1})) > 0,$$

което означава, че в A има път от s до елемент на F, тоест A не е празната матрица (по дефиниция празната матрица не е автоматна). От твърдение Б.2 следва, че

- $(\forall 1 \le i \le m)(\Pi(s, i) \neq \varnothing)$;
- $(\forall 1 \le i \le m)(\Pi(i, F) \neq \varnothing)$.

Остава да покажем, че $(\forall i \in F)(\sum_{j=1}^m a_{ij} = 0)$. Нека $i \in F$. Ако допуснем, че съществува $1 \le j \le m$, такова че $a_{ij} > 0$, то би следвало, че $a_{ij} = \xi(a, \iota^{-1}(i))$, където $\$ \not\succeq \iota^{-1}(i)$ и a е единственият елемент на $\widetilde{\Sigma} \cup \{\$\}$, такъв че $h_n(\iota^{-1}(i)a) = \iota^{-1}(j)$. Но $\$ \succeq \iota^{-1}(i)$. Стигаме до противоречие. Следователно

$$(\forall i \in F)(\sum_{j=1}^{m} a_{ij} = 0). \qquad \square$$

Твърдение Б.4. *Нека $\widetilde{\Sigma}$ е азбука, $n > 1$ и ξ е функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0,1]$. Нека $\gamma \in \{\hat{}\} \circ \widetilde{\Sigma}^\star$ и $\eta\colon \widetilde{\Sigma}^\star \circ \{\$\} \to \Pi_A(\iota(\gamma), F)$ е дефинирана за всяко $\alpha \in \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1, a_2, \dots, a_{k+1}$, такова че $\prod_{i=1}^{k+1} \xi(a_i, h_n(\gamma a_1 a_2 \dots a_{i-1})) > 0$, по следния начин*

$$\eta(\alpha) = (\iota(h_n(\gamma)), \iota(h_n(\gamma a_1)), \dots, \iota(h_n(\gamma a_1 a_2 \dots a_{k+1}))).$$

Тогава η е биекция и

$$(\forall a_1 a_2 \dots a_{k+1} \in \widetilde{\Sigma}^\star \circ \{\$\})(w(\eta(a_1 a_2 \dots a_{k+1})) = \prod_{i=1}^{k+1} \xi(a_i, h_n(\gamma a_1 a_2 \dots a_{i-1}))).$$

Доказателство. Ясно е, че η е дефинирана за всяко $\alpha \in \widetilde{\Sigma}^\star \circ \{\$\}$, $\alpha = a_1 a_2 \dots a_{k+1}$, такова че $\prod_{i=1}^{k+1} \xi(a_i, h_n(\gamma a_1 a_2 \dots a_{i-1})) > 0$, тъй като ако α е такова, то

$$(\forall 0 \leq i \leq k)(a_{\iota(h_n(\gamma a_1 \dots a_i))\iota(h_n(\gamma a_1 \dots a_{i+1}))} = \xi(a_{i+1}, h_n(\gamma a_1 \dots a_i)) > 0).$$

Ясно е, че за $\alpha \in \mathrm{Dom}(\eta)$ дължината на пътя $\eta(\alpha)$ е равна на $|\alpha|$. Поради тази причина можем лесно да съобразим, че η е инекция. Ако $\alpha, \beta \in \mathrm{Dom}(\eta)$ и $|\alpha| \neq |\beta|$, то $\eta(\alpha) \neq \eta(\beta)$, тъй като имат различна дължина. Ако $\alpha, \beta \in \mathrm{Dom}(\eta)$, $\alpha = a_1 a_2 \dots a_{k+1}$, $\beta = b_1 b_2 \dots b_{k+1}$ и $\alpha \neq \beta$, то съществува $1 \leq i \leq k$, за което $a_i \neq b_i$. Тогава $\eta(\alpha) \neq \eta(\beta)$, защото образът на $i+1$-вия елемент на $\eta(\alpha)$ относно ι^{-1} завършва на a_i, а образът на i-тия елемент на $\eta(\beta)$ относно ι^{-1} завършва на b_i.

Всеки път от $\Pi_A(\iota(\gamma), F)$ има дължина не по-малка от 1. Нека $\pi = (p_1, p_2, \dots, p_{k+2}) \in \Pi_A(\iota(\gamma), F)$. Тогава за $1 \leq i \leq k+1$ е в сила, че $a_{p_i p_{i+1}} > 0$, тоест $\xi(a_{i+1}, \iota^{-1}(p_i)) > 0$, където $a_{i+1} \in \widetilde{\Sigma} \cup \{\$\}$ е такова, че $h_n(\iota^{-1}(p_i)a_{i+1}) = \iota^{-1}(p_{i+1})$, и $\$ \not\succeq \iota^{-1}(p_i)$. Сега, тъй като $p_{k+2} \in F$, следва, че $\iota^{-1}(p_{k+2})$ завършва на \$, тоест $a_{k+2} = \$$. Тогава $a_2 \dots a_{k+2} \in \widetilde{\Sigma}^\star \circ \{\$\}$ и $\eta(\alpha) = \pi$. Така проверихме, че η е сюрекция.

Нека $\alpha \in \mathrm{Dom}(\eta)$, $\alpha = a_1 a_2 \dots a_{k+1}$. Тогава

$$\begin{aligned} w(\eta(\alpha)) &= w((\iota(h_n(\gamma)), \iota(h_n(\gamma a_1)), \dots, \iota(h_n(\gamma a_1 a_2 \dots a_{k+1})))) \\ &= \prod_{i=1}^{k+1} a_{\iota(h_n(\gamma a_1 a_2 \dots a_{i-1}))\iota(h_n(\gamma a_1 a_2 \dots a_i))} \\ &= \prod_{i=1}^{k+1} \xi(a_i \mid h_n(\gamma a_1 a_2 \dots a_{i-1})). \end{aligned}$$ □

Твърдение Б.5. *Нека $\widetilde{\Sigma}$ е азбука, $n > 1$ и ξ е функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0, 1]$. Тогава съществува n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$ тогава и само тогава, когато матрицата $A \in M_m(\mathbb{R})$ е почти стохастична и автоматна.*

Доказателство.

⇒ От твърдение Б.3 следва, че A е автоматна. Остава да покажем, че тя е почти стохастична и по-точно, че

$$(\forall 1 \leq i \leq m)(i \notin F \implies \sum_{j=1}^{m} a_{ij} = 1).$$

Нека $i \in \{1, 2, \dots, m\} \setminus F$. Ясно е, че съществува път в A от s до $f \in F$, минаващ през i. Нека това да бъде пътят $\pi = (p_1, p_2, \dots, p_{k+1})$, където $p_1 = s$, $p_r = i$ за някое $1 \leq r \leq k$ и $p_{k+1} = f$. Теглото на пътя е

$$\omega(\pi) = \prod_{i=2}^{k+1} \xi(a_i, \iota^{-1}(p_{i-1})) > 0,$$

където a_i за $2 \leq i \leq k+1$ е единственият елемент на $\widetilde{\Sigma} \cup \{\$\}$, такъв че $h_n(\iota^{-1}(p_{i-1})a_i) = \iota^{-1}(p_i)$. Тогава, тъй като $\iota^{-1}(s) = \hat{}$ и $\$ \succeq \iota^{-1}(f)$, следва, че $\hat{}a_2 a_3 \dots a_{k+1} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$ и

$$\omega(\pi) = \prod_{i=2}^{k+1} P(B_{\hat{}a_2 \dots a_i} \mid B_{\hat{}a_2 \dots a_{i-1}}) > 0.$$

Оттук следва, че $P(B_{\hat{}a_2\dots a_r}) > 0$, което от своя страна влече, че

$$\sum_{j=1}^{m} a_{ij} = \sum_{a\in\widetilde{\Sigma}\cup\{\$\}} \xi(a, h_n(\hat{}a_2\dots a_r)) = \sum_{a\in\widetilde{\Sigma}\cup\{\$\}} P(B_{\hat{}a_2\dots a_r a} \mid B_{\hat{}a_2\dots a_r}) = 1.$$

$\Leftarrow$ Да разгледаме изброимо адитивната функция $P\colon \mathcal{P}(\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}) \to [0,1]$, дефинирана за $\alpha \in \{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}$, $\alpha = a_1a_2\dots a_{k+2}$ така

$$P(\{\alpha\}) = \prod_{i=2}^{k+2} \xi(a_i, h_n(a_1a_2\dots a_{i-1})).$$

За да докажем, че P е езиков модел е достатъчно да покажем, че

$$\sum_{\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}} P(\{\alpha\}) = 1.$$

От твърдение Б.4 за $\gamma = \hat{}$ и твърдение А.6 следва, че

$$\begin{aligned}
\sum_{\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}} P(\{\alpha\}) &= \sum_{\substack{\alpha\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}:\\ P(\{\alpha\})>0}} P(\{\alpha\}) \\
&= \sum_{i=2}^{\infty} \sum_{\substack{a_1a_2\dots a_i\in\{\hat{}\}\circ\widetilde{\Sigma}\circ\{\$\}:\\ P(\{a_1a_2\dots a_i\})>0}} \prod_{j=2}^{i} \xi(a_i, h_n(a_1a_2\dots a_{i-1})) \\
&= \sum_{\pi\in\Pi_A(s,F)} \omega(\pi) \\
&= 1.
\end{aligned}$$

Сега ще покажем, че за всяко $a \in \widetilde{\Sigma}\cup\{\$\}$ и всяко $\alpha \in \{\hat{}\}\circ\widetilde{\Sigma}^\star$, такова че $P(B_\alpha) > 0$, е в сила, че

$$P(B_{\alpha a} \mid B_\alpha) = \xi(a, h_n(\alpha)),$$

от което следва, че P е n-грамен езиков модел и $\xi \in \Xi_P$. Наистина нека $a \in \widetilde{\Sigma}\cup\{\$\}$ и $\alpha \in \{\hat{}\}\circ\widetilde{\Sigma}^\star$, $\alpha = a_1a_2\dots a_{k+1}$ е такова, че $P(B_\alpha) > 0$. Тогава

$$P(B_{\alpha a} \mid B_\alpha) = \frac{P(B_{\alpha a})}{P(B_\alpha)} = \frac{\sum_{\substack{\beta\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}:\\ \alpha a\preceq\beta}} P(\{\beta\})}{\sum_{\substack{\beta\in\{\hat{}\}\circ\widetilde{\Sigma}^\star\circ\{\$\}:\\ \alpha\preceq\beta}} P(\{\beta\})}.$$

Нека

$$\begin{aligned}
X &= \prod_{i=2}^{k+1} \xi(a_i, h_n(a_1a_2\dots a_{i-1})), \\
Y &= \sum_{i=0}^{\infty} \sum_{\substack{b_1b_2\dots b_{i+1}\in\\ \widetilde{\Sigma}^\star\circ\{\$\}}} \left(\prod_{j=1}^{i} \xi(b_j, h_n(\alpha b_1\dots b_{j-1}))\right) \xi(\$, h_n(\alpha b_1\dots b_i)).
\end{aligned}$$

Тогава числителят е равен на

$$\begin{cases} X\xi(a, h_n(\alpha)), & \text{ако } a = \$ \\ X\xi(a, h_n(\alpha))Y, & \text{иначе} \end{cases}$$

а знаменателят е равен на XY. Сега, използвайки твърдение Б.4 за $\gamma = \alpha$ и твърдение А.6, получаваме, че

$$Y = \sum_{\pi \in \Pi_A(\iota(\alpha), F)} \omega(\pi) = 1,$$

тоест

$$P(B_{\alpha a} \mid B_\alpha) = \xi(a, h_n(\alpha)). \qquad \square$$

В заключение ще докажем необходимото и достатъчно условие функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0, 1]$ да определя n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, формулирано в твърдение 1.3 и отразяващо същината на наученото от досега доказаните твърдения.

Твърдение Б.6. *Нека $\widetilde{\Sigma}$ е азбука, $n \in \mathbb{N}^+$ и ξ е функция от $(\widetilde{\Sigma} \cup \{\$\}) \times \{h_n(\alpha) \mid \alpha \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\}$ в $[0, 1]$. Тогава съществува n-грамен езиков модел P над $\{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\}$, такъв че $\xi \in \Xi_P$, тогава и само тогава, когато*

- $(\exists a_1 a_2 \dots a_{m+2} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})(\prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1}) > 0)$;
- $(\forall \alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\})(\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \alpha) = 1)$.

Доказателство. За $n = 1$ твърдението следва от твърдение Б.1, тъй като

$$(\exists a_1 a_2 \dots a_{m+2} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})(\prod_{i=2}^{m+2} \xi(a_i, h_1(a_1 a_2 \dots a_{i-1}) > 0) \iff \xi(\$, \varepsilon) > 0$$

$$(\forall \alpha \in \{h_1(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\})(\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \alpha) = 1) \iff \sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \varepsilon) = 1.$$

За $n > 1$ правата посока следва директно от твърдение Б.5. За обратната посока, съгласно твърдение Б.5, е достатъчно да проверим, че $A \in M_m(\mathbb{R})$ е автоматна и почти стохастична. Тъй като

$$(\exists a_1 a_2 \dots a_{m+2} \in \{\hat{}\} \circ \widetilde{\Sigma}^\star \circ \{\$\})(\prod_{i=2}^{m+2} \xi(a_i, h_n(a_1 a_2 \dots a_{i-1}) > 0),$$

можем да повторим доказателството на твърдение Б.3 и да заключим, че A е автоматна. За да докажем, че A е почти стохастична, остава да покажем, че

$$(\forall 1 \leq i \leq m)(i \notin F \implies \sum_{j=1}^{m} a_{ij} = 1).$$

Но това следва от условието

$$(\forall \alpha \in \{h_n(\beta) \mid \beta \in \{\hat{}\} \circ \widetilde{\Sigma}^\star\})(\sum_{a \in \widetilde{\Sigma} \cup \{\$\}} \xi(a, \alpha) = 1). \qquad \square$$

Библиография

Aho, Alfred V., и Margaret J. Corasick. 1975. „Efficient string matching: An aid to bibliographic search". 18 (6): 333—340.

Allauzen, Cyril, Mehryar Mohri и Brian Roark. 2003. „Generalized algorithms for constructing statistical language models". В *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics,* 1:40—47.

Béal, Marie-Pierre, и Maxime Crochemore. 2008. „Minimizing incomplete automata". В *FSMNLP 2008.*

Björklund, Henrik, Johanna Björklund и Niklas Zechner. 2014. „Compression of finite-state automata through failure transitions". *Theoretical Computer Science* 557:87—100.

Chen, Stanley F., и Joshua Goodman. 1998. *An empirical study of smoothing techniques for language modeling.* Технически доклад TR-10-98. Computer Science Group, Harvard University.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest и Clifford Stein. 2009. *Introduction to algorithms.* 3-е издание. The MIT Press.

E. Knuth, Donald, James H. Morris Jr и Vaughan Pratt. 1977. „Fast Pattern Matching in Strings". *SIAM J. Comput.* 6:323—350.

Hopcroft, John E. 1971. *An $n \log n$ algorithm for minimizing states in a finite automaton.* Технически доклад. Stanford University.

Hopcroft, John E., и Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages and computation.* 1-е издание. Addison-Wesley Publishing Company.

Kneser, Reinhard, и Hermann Ney. 1995. „Improved backing-off for m-gram language modeling". В *1995 International Conference on Acoustics, Speech, and Signal Processing.*

Mihov, Stoyan, и Klaus U. Schulz. 2018 (in press). *Finite-State Techniques: Automata, Transducers and Bimachines.* Cambridge University Press.

Mohri, Mehryar. 1997. „Finite-state transducers in language and speech processing". *Computational Linguistics* 23 (2): 269—311.

Mohri, Mehryar. 2000. „Minimization algorithms for sequential transducers". *Theoretical Computer Science* 234 (1): 177—201.

Ney, Hermann, Sven Martin и Frank Wessel. 1997. В *Corpus-Based Methods in Language and Speech Processing,* под редакцията на Steve Young и Gerrit Bloothooft, 174—207. Dordrecht: Springer Netherlands.

Stolcke, Andreas. 2002. „SRILM - an extensible language modeling toolkit". В *INTERSPEECH.*

Stolcke, Andreas, Jing Zheng, Wen Fu Wang и Victor Abrash. 2011. „SRILM at Sixteen: Update and Outlook". В *Proceedings IEEE Automatic Speech Recognition and Understanding Workshop.*