Uniwersytet im. Adama Mickiewicza
Wydział Matematyki i Informatyki

Paweł Skórzewski
nr albumu: 301654

Gramatyki i automaty probabilistyczne

Praca magisterska na kierunku:
matematyka

Promotor:
prof. dr hab. Wojciech Buszkowski

Poznań 2010

Oświadczenie

Poznań, dnia .....

Ja, niżej podpisany Paweł Skórzewski, student Wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu, oświadczam, że przedkładaną pracę dyplomową pt. *Gramatyki i automaty probabilistyczne* napisałem samodzielnie. Oznacza to, że przy pisaniu pracy, poza niezbędnymi konsultacjami, nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opracowania rozprawy lub jej części innym osobom ani nie odpisywałem tej rozprawy lub jej części od innych osób.

Oświadczam również, że egzemplarz pracy dyplomowej w formie wydruku komputerowego jest zgodny z egzemplarzem pracy dyplomowej w formie elektronicznej.

Jednocześnie przyjmuję do wiadomości, że gdyby powyższe oświadczenie okazało się nieprawdziwe, decyzja o wydaniu mi dyplomu zostanie cofnięta.

.....

Spis treści

Rozdział 1

Wstęp

Wiek dwudziesty był wiekiem wielkiego rozwoju matematyki. Wśród licznych dziedzin, które rozwijały się niezwykle dynamicznie, znalazła się logika, w związku z debatą na temat podstaw matematyki. Wynalazek komputera przyczynił się z kolei do powstania informatyki teoretycznej. Ważnymi gałęziami informatyki teoretycznej są teoria automatów i teoria języków formalnych — powstałe w połowie ubiegłego wieku dziedziny, które mają korzenie w logice i szerokie zastosowania w informatyce. Narzędzia tych dwóch blisko związanych ze sobą teorii wykorzystywane są do modelowania maszyn liczących; do opisu języków: zarówno sztucznych, jak języki programowania, jak i naturalnych; przy ich pomocy tworzy się wszechstronne oprogramowanie: od kompilatorów po aplikacje do rozpoznawania mowy czy tłumaczenia automatycznego.

Użycie w teorii automatów i języków formalnych narzędzi innej dynamicznie rozwijającej się dyscypliny matematyki dwudziestego wieku — rachunku prawdopodobieństwa — otwiera nowe perspektywy dla rozwoju tych dziedzin i ich zastosowań; umożliwia stosowanie nowych metod i nowych narzędzi, pozwalając szerzej wykorzystać je w informatyce.

Niniejsza praca stanowi krótki przegląd najważniejszych pojęć i wybranych zagadnień dotyczących gramatyk i automatów probabilistycznych.

Rozdział 2 zawiera podstawowe pojęcia z różnych dziedzin matematyki, w szczególności teorii mnogości, rachunku prawdopodobieństwa, kombinatoryki i teorii języków formalnych, które są wykorzystywane w niniejszej pracy. Omówiono pojęcie relacji, pojęcie antyłańcucha; prawdopodobieństwo i jego własności, rozkład prawdopodobieństwa, prawdopodobieństwo warunkowe, niezależność zdarzeń, pojęcie zmiennej losowej i jej rozkładu prawdopodobieństwa, niezależność zmiennych losowych; grafy i drzewa; symbole, alfabety, łańcuchy, języki, automaty, gramatyki, klasyfikację języków i gramatyk, wyprowadzenia i drzewa.

Rozdział 3 poświęcony jest automatom probabilistycznym i łańcuchom Markowa. Przedstawiłem w nim definicje i podstawowe własności dotyczące probabilistycznych automatów skończonych oraz języków stochastycznych,

w tym własne dowody zależności między językami 0-stochastycznymi i językami regularnymi. Omówiłem też modele Markowa oraz N-gramowe modele języka, a także metody ich budowania na podstawie danych statystycznych i zastosowania w analizie języka naturalnego.

Rozdział 4 omawia probabilistyczne gramatyki bezkontekstowe. Przedstawiłem pojęcia prawdopodobieństw wyprowadzenia, drzewa rozkładu i łańcucha. Wkład własny stanowi sformułowanie i dowiedzenie twierdzeń o sumie prawdopodobieństw wyprowadzeń (twierdzenia 4.3, 4.4 i 4.5). Omówiłem sposoby efektywnego obliczania prawdopodobieństw łańcuchów: algorytmy wewnętrzny i zewnętrzny; a także pojęcia: prawdopodobieństwa wewnętrznego i zewnętrznego oraz drzewa Viterbiego; przedstawiłem też pokrótce ideę uczenia gramatyki. Rozdział kończy prezentacja algorytmu CYK jako przykładu algorytmu parsowania gramatyk probabilistycznych wraz z własnym dowodem jego poprawności.

Rozdział 2

Podstawowe pojęcia

2.1. Podstawowe pojęcia z zakresu teorii mnogości

Definicja 2.1 (zbiór potęgowy)**.** Zbiór $\mathcal{P}(X)$ złożony ze wszystkich podzbiorów zbioru X nazywamy *zbiorem potęgowym* zbioru X.

Definicja 2.2 (relacja)**.** *Relacją* na zbiorze X nazywamy dowolny podzbiór iloczynu kartezjańskiego $X \times X$. Jeżeli $\mathcal{R} \subseteq X \times X$ jest relacją na zbiorze X, to zależność $(x, y) \in \mathcal{R}$ zapisujemy często jako $x\mathcal{R}y$.

Definicja 2.3 (zwrotność)**.** Relację $\mathcal{R}$ na zbiorze X nazywamy *zwrotną*, jeżeli $x\mathcal{R}x$ dla każdego $x \in X$.

Definicja 2.4 (przechodniość)**.** Relację $\mathcal{R}$ na zbiorze X nazywamy *przechodnią*, jeżeli dla dowolnych $x, y, z \in X$: jeśli $x\mathcal{R}y$ i $y\mathcal{R}z$, to $x\mathcal{R}z$.

Definicja 2.5 (domknięcie zwrotne i przechodnie)**.** Jeżeli $\mathcal{R}$ jest relacją na zbiorze X, to *domknięciem zwrotnym i przechodnim* relacji $\mathcal{R}$ nazywamy najmniejszą (ze względu na inkluzję) relację $\mathcal{R}^* \supseteq \mathcal{R}$, która jest zwrotna i przechodnia.

Definicja 2.6 (częściowy porządek)**.** Relację $\preceq$ na zbiorze X nazywamy *częściowym porządkiem*, jeżeli jest zwrotna i przechodnia oraz antysymetryczna, tj. jeśli $x \preceq y$ i $y \preceq x$, to $x = y$, dla dowolnych $x, y \in X$,

Mówimy wtedy, że zbiór X jest *częściowo uporządkowany* przez relację $\preceq$.

Definicja 2.7 (antyłańcuch)**.** Niech X będzie zbiorem częściowo uporządkowanym przez relację $\preceq$. Wówczas *antyłańcuchem* nazywamy każdy podzbiór $A \subseteq X$ taki, że dla dowolnych różnych $x, y \in X$ nie zachodzi ani $x \preceq y$, ani $y \preceq x$.

Definicja 2.8 (maksymalny antyłańcuch)**.** Niech X będzie zbiorem częściowo uporządkowanym przez relację $\preceq$. Antyłańcuch $A \subseteq X$ nazywamy *maksymalnym*, jeżeli dla dowolnego $x \in X \setminus A$ zbiór $A \cup \{x\}$ nie jest antyłańcuchem w X.

2.2. Podstawowe pojęcia i definicje związane z rachunkiem prawdopodobieństwa[1]

2.2.1. Pojęcie prawdopodobieństwa

Definicja 2.9 (ciało przeliczalnie addytywne)**.** Niech $\mathcal{F}$ będzie rodziną podzbiorów zbioru Ω, która spełnia warunki:

(1) $\Omega \in \mathcal{F}$,
(2) jeżeli $A \in \mathcal{F}$, to $\Omega \setminus A \in \mathcal{F}$ (komplementatywność),
(3) jeżeli $A_1, A_2, \ldots \in \mathcal{F}$ jest przeliczalnym ciągiem zbiorów z $\mathcal{F}$, to

$$\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$$

(przeliczalna addytywność).

Wówczas rodzinę $\mathcal{F}$ nazywamy *ciałem przeliczalnie addytywnym* albo *σ-ciałem.*

Twierdzenie 2.1 (własności ciał przeliczalnie addytywnych)**.** *Każde ciało przeliczalnie addytywne $\mathcal{F}$ podzbiorów zbioru Ω posiada następujące własności:*

(1) $\emptyset \in \mathcal{F}$.
(2) Jeżeli $A_1, A_2, \ldots \in \mathcal{F}$ *jest przeliczalnym ciągiem zbiorów z* $\mathcal{F}$*, to*

$$\bigcap_{n=1}^{\infty} A_n \in \mathcal{F}.$$

(3) Jeżeli $A, B \in \mathcal{F}$*, to* $A \setminus B \in \mathcal{F}$.

Dowód. (1) $\emptyset = \Omega \setminus \Omega \in \mathcal{F}$.
(2) Korzystając z praw De Morgana, otrzymujemy:

$$\bigcap_{n=1}^{\infty} A_n = \bigcap_{n=1}^{\infty} (\Omega \setminus (\Omega \setminus A_n)) = \Omega \setminus \bigcup_{n=1}^{\infty} (\Omega \setminus A_n) \in \mathcal{F}.$$

(3) $A \setminus B = A \cap (\Omega \setminus B) \in \mathcal{F}$.

□

Możemy zatem powiedzieć, że ciało przeliczalnie addytywne jest rodziną zbiorów zamkniętą na co najwyżej przeliczalne operacje mnogościowe.

Zbiór potęgowy $\mathcal{P}(\Omega)$ wszystkich podzbiorów zbioru Ω jest ciałem przeliczalnie addytywnym. Jest to największe (ze względu na relację inkluzji) ciało przeliczalnie addytywne podzbiorów zbioru Ω.

[1] Na podstawie [14].

Definicja 2.10 (przestrzeń probabilistyczna)**.** *Przestrzeń probabilistyczna* to trójka uporządkowana $(\Omega, \mathcal{F}, \mathbf{P})$, gdzie:

— Ω jest dowolnym zbiorem, nazywanym *przestrzenią zdarzeń elementarnych*,

— $\mathcal{F}$ jest pewnym ciałem przeliczalnie addytywnym podzbiorów zbioru Ω, zbiór $\mathcal{F}$ nazywany jest *zbiorem zdarzeń*, a jego elementy *zdarzeniami*.

— funkcja $\mathbf{P}\colon \mathcal{F} \to \mathbb{R}$, nazywana *prawdopodobieństwem*, spełnia następujące aksjomaty:

(1) $0 \leqslant \mathbf{P}(A) \leqslant 1$ dla każdego $A \in \mathcal{F}$,

(2) $\mathbf{P}(\Omega) = 1$,

(3) dla każdego ciągu $A_1, A_2, \ldots \in \mathcal{F}$ parami rozłącznych zdarzeń (tj. $A_i \cap A_j = \emptyset$ dla $i \neq j$) zachodzi równość

$$\mathbf{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbf{P}\left(A_n\right)$$

(przeliczalna addytywność).

Zdarzenie Ω nazywamy *zdarzeniem pewnym*. Zdarzenie $\emptyset$ nazywamy *zdarzeniem niemożliwym*.

Twierdzenie 2.2 (własności prawdopodobieństwa)**.** *Każda przestrzeń probabilistyczna $(\Omega, \mathcal{F}, \mathbf{P})$ posiada następujące własności:*

(1) $\mathbf{P}(\emptyset) = 0$.

(2) $\mathbf{P}(\Omega \setminus A) = 1 - \mathbf{P}(A)$ *dla* $A \in \mathcal{F}$.

Dowód.

(1) $\mathbf{P}(\emptyset) = \mathbf{P}\left(\bigcup_{n=1}^{\infty} \emptyset\right) = \sum_{n=1}^{\infty} \mathbf{P}(\emptyset)$, zatem $\mathbf{P}(\emptyset) = 0$.

(2) $1 = \mathbf{P}(\Omega) = \mathbf{P}\left(A \cup (\Omega \setminus A)\right) = \mathbf{P}(A) + \mathbf{P}(\Omega \setminus A)$, zatem $\mathbf{P}(\Omega \setminus A) = 1 - \mathbf{P}(A)$. □

2.2.2. Rozkłady prawdopodobieństwa

Twierdzenie 2.3 (rozkład prawdopodobieństwa na co najwyżej przeliczalnej przestrzeni zdarzeń elementarnych)**.** *Niech $\Omega = \{e_1, e_2, \ldots\}$ będzie zbiorem co najwyżej przeliczalnym. Niech $\mathcal{F} = \mathcal{P}(\Omega)$. Niech $p\colon \Omega \to \mathbb{R}$ będzie funkcją taką, że $p(e) \geqslant 0$ dla każdego $e \in \Omega$ oraz*

$$\sum_{e \in \Omega} p(e) = 1.$$

Określmy funkcję $\mathbf{P}\colon \mathcal{F} \to \mathbb{R}$ następująco:

$$\mathbf{P}(A) := \sum_{e \in A} p(e).$$

Wówczas $(\Omega, \mathcal{F}, \mathbf{P})$ jest przestrzenią probabilistyczną.

Dowód. Funkcja $\mathbf{P}$ spełnia warunek (1) definicji 2.10, ponieważ

$$0 \leqslant p(e) \leqslant \mathbf{P}(A) = \sum_{e \in A} p(e) \leqslant \sum_{e \in \Omega} p(e) = 1 \quad \text{dla dowolnego } e \in A.$$

Warunek (2) spełniony jest, ponieważ

$$\mathbf{P}(\Omega) = \sum_{e \in \Omega} p(e) = 1.$$

Funkcja $\mathbf{P}$ spełnia również warunek (3), ponieważ dla parami rozłącznych zbiorów $A_1, A_2, \ldots$ zachodzą równości

$$\mathbf{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{e \in \bigcup_{n=1}^{\infty} A_n} p(e) = \sum_{n=1}^{\infty} \sum_{e \in A_n} p(e) = \sum_{n=1}^{\infty} \mathbf{P}\left(A_n\right).$$

□

Funkcję p z powyższego twierdzenia nazywa się *rozkładem prawdopodobieństwa* na (co najwyżej przeliczalnej) przestrzeni Ω.

Definicja 2.11 (ciało zbiorów borelowskich)**.** *Ciałem zbiorów borelowskich na prostej* $\mathbb{R}$ nazywamy najmniejsze (ze względu na inkluzję) ciało przeliczalnie addytywne zawierające przedziały postaci $(-\infty, a)$, $a \in \mathbb{R}$ i oznaczamy przez $\mathcal{B}(\mathbb{R})$.

Definicja 2.12 (rozkład prawdopodobieństwa na prostej)**.** Każdą funkcję $\mathbf{P}$ taką, że $(\mathbb{R}, \mathcal{B}(\mathbb{R}), \mathbf{P})$ jest przestrzenią probabilistyczną, nazywamy *rozkładem prawdopodobieństwa na prostej.*

Definicja 2.13 (dystrybuanta)**.** Jeżeli $\mathbf{P}$ jest rozkładem prawdopodobieństwa na prostej, to funkcję $F\colon \mathbb{R} \to \mathbb{R}$ określoną wzorem

$$F(x) := \mathbf{P}((-\infty, x))$$

nazywamy *dystrybuantą* albo *funkcją rozkładu prawdopodobieństwa* $\mathbf{P}$.

Twierdzenie 2.4. *Dystrybuanta* $F\colon \mathbb{R} \to \mathbb{R}$ *dowolnego rozkładu prawdopodobieństwa na prostej spełnia warunki:*

(1) F *jest niemalejąca,*
(2) $\lim_{x \to -\infty} F(x) = 0$ *oraz* $\lim_{x \to \infty} F(x) = 1$*,*
(3) F *jest lewostronnie ciągła, tj.* $\lim_{x \to a-} F(x) = F(a)$ *dla każdego* $a \in \mathbb{R}$*,*

Twierdzenie 2.5. *Każda funkcja spełniająca warunki (1)-(3) z twierdzenia 2.4 jest dystrybuantą jakiegoś rozkładu prawdopodobieństwa na prostej.*

Dowód. Dowody obu powyższych twierdzeń są przedstawione w [14]. □

Definicja 2.14 (skok)**.** Niech $F\colon \mathbb{R} \to \mathbb{R}$. Jeżeli $\mathbf{P}(\{a\}) = \lim_{x\to a+} F(x) - F(a) > 0$, to mówimy, ze funkcja F ma w punkcie a skok równy $\mathbf{P}(\{a\})$.

Twierdzenie 2.6. *Zbiór punktów skoku dystrybuanty jest co najwyżej przeliczalny.*

Dowód. Dowód przedstawiony jest w [14]. □

Definicja 2.15 (dyskretny rozkład prawdopodobieństwa)**.** Rozkład prawdopodobieństwa $\mathbf{P}$ na prostej nazywamy *dyskretnym*, jeżeli suma wszystkich skoków jego dystrybuanty wynosi 1.

2.2.3. Prawdopodobieństwo warunkowe, niezależność zdarzeń

Definicja 2.16 (prawdopodobieństwo warunkowe)**.** Niech $(\Omega, \mathcal{F}, \mathbf{P})$ będzie przestrzenią probabilistyczną. Jeżeli $A, B \in \mathcal{F}$, $\mathbf{P}(B) > 0$, to *prawdopodobieństwo warunkowe* zdarzenia A pod warunkiem, że zaszło zdarzenie B, definiujemy jako iloraz $\mathbf{P}(A \cap B)$ przez $\mathbf{P}(B)$:

$$\mathbf{P}(A|B) := \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(B)}.$$

Twierdzenie 2.7. *Niech $(\Omega, \mathcal{F}, \mathbf{P})$ będzie przestrzenią probabilistyczną, zaś $B \in \mathcal{F}$ zdarzeniem takim, że $\mathbf{P}(B) > 0$. Niech $\mathbf{P}_B\colon \mathcal{F} \to \mathbb{R}$ będzie funkcją określoną następująco: $\mathbf{P}_B(A) := \mathbf{P}(A|B)$. Wówczas $(\Omega, \mathcal{F}, \mathbf{P}_B)$ jest przestrzenią probabilistyczną.*

Dowód. Funkcja $\mathbf{P}_B$ spełnia warunki definicji 2.10, ponieważ:

(1) $\mathbf{P}_B(A) = \frac{\mathbf{P}(A\cap B)}{\mathbf{P}(B)} \geqslant \mathbf{P}(A \cap B) \geqslant 0$.
(2) $\mathbf{P}_B(\Omega) = \frac{\mathbf{P}(\Omega\cap B)}{\mathbf{P}(B)} = \frac{\mathbf{P}(B)}{\mathbf{P}(B)} = 1$.
(3) Jeżeli $A_1, A_2, \ldots$ jest ciągiem parami rozłącznych zdarzeń, to $A_1 \cap B, A_2 \cap B, \ldots$ jest również ciągiem parami rozłącznych zdarzeń. Wówczas

$$\mathbf{P}_B\left(\bigcup_{n=1}^{\infty} A_n\right) = \frac{\mathbf{P}\left(\left(\bigcup_{n=1}^{\infty} A_n\right) \cap B\right)}{\mathbf{P}(B)} = \frac{\mathbf{P}\left(\bigcup_{n=1}^{\infty}(A_n \cap B)\right)}{\mathbf{P}(B)} =$$
$$= \frac{1}{\mathbf{P}(B)} \sum_{n=1}^{\infty} \mathbf{P}(A_n \cap B) = \sum_{n=1}^{\infty} \frac{\mathbf{P}(A_n \cap B)}{\mathbf{P}(B)} = \sum_{n=1}^{\infty} \mathbf{P}_B(A).$$

□

Twierdzenie 2.8 (prawdopodobieństwo całkowite)**.** *Niech $(\Omega, \mathcal{F}, \mathbf{P})$ będzie przestrzenią probabilistyczną. Jeżeli $A_1, A_2, \ldots \in \mathcal{F}$ jest co najwyżej przeliczalnym ciągiem parami rozłącznych zdarzeń takich, że $\bigcup_{i=1}^{\infty} A_i = \Omega$ oraz $\mathbf{P}(A_i) > 0$ dla wszystkich $i = 1, 2, \ldots$, to dla każdego zdarzenia $A \in \mathcal{F}$ zachodzi następujący* wzór na prawdopodobieństwo całkowite*:*

$$\mathbf{P}(A) = \sum_{i=1}^{\infty} \mathbf{P}(A|A_i)\mathbf{P}(A_i).$$

Dowód. Zdarzenia $A \cap A_1, A \cap A_2, \ldots$ są rozłaczne, więc

$$\mathbf{P}(A) = \mathbf{P}(A \cap \Omega) = \mathbf{P}\left(\bigcup_{i=1}^{\infty}(A \cap A_i)\right) = \\ = \sum_{i=1}^{\infty}\mathbf{P}(A \cap A_i) = \sum_{i=1}^{\infty}\frac{\mathbf{P}(A \cap A_i)}{\mathbf{P}(A_i)}\mathbf{P}(A_i) = \sum_{i=1}^{\infty}\mathbf{P}(A|A_i)\mathbf{P}(A_i).$$

□

Twierdzenie 2.9 (twierdzenie Bayesa)**.** *Niech* $(\Omega, \mathcal{F}, \mathbf{P})$ *będzie przestrzenią probabilistyczną. Dla każdych dwóch zdarzeń* $A, B \in \mathcal{F}$ *takich, że* $\mathbf{P}(A) > 0$ *i* $\mathbf{P}(B) > 0$*, zachodzi zależność*

$$\mathbf{P}(B|A) = \frac{\mathbf{P}(A|B)\mathbf{P}(B)}{\mathbf{P}(A)}.$$

Dowód. Teza twierdzenia wynika z następujących równości:

$$\mathbf{P}(B|A) = \frac{\mathbf{P}(B \cap A)}{\mathbf{P}(A)} = \frac{\frac{\mathbf{P}(B \cap A)}{\mathbf{P}(B)}\mathbf{P}(B)}{\mathbf{P}(A)} = \frac{\mathbf{P}(A|B)\mathbf{P}(B)}{\mathbf{P}(A)}.$$

□

Definicja 2.17 (niezależność zdarzeń)**.** Niech $(\Omega, \mathcal{F}, \mathbf{P})$ będzie przestrzenią probabilistyczną. Zdarzenia $A, B \in \mathcal{F}$ nazywamy *zdarzeniami niezależnymi*, jeżeli prawdopodobieństwo ich iloczynu jest iloczynem ich prawdopodobieństw, czyli

$$\mathbf{P}(A \cap B) = \mathbf{P}(A) \cdot \mathbf{P}(B).$$

2.2.4. Zmienne losowe

Definicja 2.18 (zmienna losowa)**.** Jeżeli $(\Omega, \mathcal{F}, \mathbf{P})$ jest przestrzenią probabilistyczną, to *zmienną losową* nazywamy każdą funkcję

$$X\colon \Omega \to \mathbb{R}$$

taką, że dla każdej liczby $a \in \mathbb{R}$ zachodzi

$$\{e \in \Omega\colon X(e) < a\} \in \mathcal{F}$$

(czyli $X^{-1}(B) \in \mathcal{F}$ dla dowolnego zbioru borelowskiego $B \in \mathcal{B}(\mathbb{R})$).

Zamiast

$$\mathbf{P}(\{\omega \in \Omega\colon X(\omega) < a\})$$

będziemy często pisać

$$\mathbf{P}(X < a).$$

Podobnie należy rozumieć notacje: $\mathbf{P}(X > a)$, $\mathbf{P}(X = a)$, $\mathbf{P}(X \in A)$ itp.

Definicja 2.19 (dystrybuanta zmiennej losowej)**.** *Dystrybuantą zmiennej losowej* $X\colon \Omega \to \mathbb{R}$ nazywamy funkcję $F_X\colon \mathbb{R} \to \mathbb{R}$ określoną jako

$$F_X(x) := \mathbf{P}(X < x) = \mathbf{P}(\{\omega \in \Omega\colon X(\omega) < x\}).$$

Definicja 2.20 (rozkład prawdopodobieństwa zmiennej losowej)**.** *Rozkładem prawdopodobieństwa zmiennej losowej* $X\colon \Omega \to \mathbb{R}$ nazywamy rozkład prawdopodobieństwa na prostej, którego dystrybuantą jest F_X.

Definicja 2.21 (niezależność zmiennych losowych)**.** Niech $(\Omega, \mathcal{F}, \mathbf{P})$ będzie przestrzenią probabilistyczną. Zmienne losowe $X, Y\colon \Omega \to \mathbb{R}$ nazywamy *niezależnymi*, jeżeli dla dowolnych zbiorów borelowskich $A, B \in \mathcal{B}(\mathbb{R})$ zachodzi równość:

$$\mathbf{P}(X \in A, Y \in B) = \mathbf{P}(X \in A)\,\mathbf{P}(Y \in B).$$

Definicja 2.22 (dyskretna zmienna losowa)**.** Zmienną losową nazywamy *dyskretną*, jeżeli jej rozkład prawdopodobieństwa jest dyskretny.

Twierdzenie 2.10. *Zbiór wszystkich wartości dyskretnej zmiennej losowej, które przyjmuje z dodatnim prawdopodobieństwem, jest co najwyżej przeliczalny.*

Dowód. Gdyby dyskretna zmienna losowa przyjmowała nieprzeliczalnie wiele wartości z niezerowym prawdopodobieństwem, to suma wartości skoków jej dystrybuanty nie mogłaby być skończona. □

Z tego powodu możemy uogólnić pojęcie dyskretnej zmiennej losowej i mianem tym określać każdą funkcję określoną na przestrzeni zdarzeń elementarnych o wartościach w zbiorze co najwyżej przeliczalnym A taką, że $X^{-1}(B) \in \mathcal{F}$ dla dowolnego $B \subseteq A$.

2.3. Podstawowe pojęcia i definicje z zakresu teorii grafów[2]

Definicja 2.23 (graf)**.** *Grafem (prostym)* nazywamy parę uporządkowaną $\Gamma = (U, E)$, gdzie U jest skończonym zbiorem *wierzchołków*, zaś $E \subseteq \{\{u, v\}\colon u, v \in U\}$ jest skończonym zbiorem *krawędzi*, czyli (nieuporządkowanych) par wierzchołków.

O krawędzi $\{u, v\}$ mówimy, że *łączy* wierzchołki u i v. Mówimy wówczas też, że wierzchołki u i v są *sąsiednie* albo *sąsiadują ze sobą*.

[2]Na podstawie [13] i [2].

Definicja 2.24 (graf skierowany)**.** *Grafem skierowanym* albo *digrafem* nazywamy parę uporządkowaną $\Gamma = (U, E)$, gdzie U jest skończonym zbiorem *wierzchołków*, zaś $E \subseteq U \times U = \{(u, v)\colon u, v \in U\}$ jest skończonym zbiorem *łuków*, czyli uporządkowanych par wierzchołków.

Definicja 2.25 (poprzednik, następnik)**.** Jeżeli (u, v) jest łukiem grafu skierowanego, to wierzchołek u nazywamy *poprzednikiem* wierzchołka v, a wierzchołek v nazywamy *następnikiem* wierzchołka u.

Definicja 2.26 (podgraf)**.** *Podgrafem* grafu $\Gamma = (U, E)$ nazywamy każdy taki graf $\Gamma' = (U', E')$, że $U' \subseteq U$ oraz $E' \subseteq E$.

Analogicznie definiujemy podgraf grafu skierowanego.

Definicja 2.27 (trasa)**.** Skończony ciąg krawędzi $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$ (być może zerowej długości) grafu $\Gamma = (U, E)$ nazywamy *trasą* z wierzchołka v_0 do wierzchołka v_n o długości n. Wierzchołek v_0 nazywamy *początkiem*, a wierzchołek v_n — *końcem* trasy.

Analogicznie definiujemy trasę w grafie skierowanym (zastępując w niniejszej definicji krawędzie łukami).

Definicja 2.28 (ścieżka)**.** *Ścieżką* nazywamy trasę, której wszystkie krawędzie są różne.

Definicja 2.29 (ścieżka zamknięta)**.** Jeżeli początek i koniec ścieżki są takie same, to taką ścieżkę nazywamy *zamkniętą*.

Definicja 2.30 (cykl)**.** *Cyklem* nazywamy każdą ścieżkę zamkniętą, która posiada co najmniej jedną krawędź.

Definicja 2.31 (droga)**.** *Drogą* nazywamy ścieżkę, której wszystkie wierzchołki, oprócz być może początku i końca, są parami różne.

Definicja 2.32 (graf spójny)**.** Graf (prosty) nazywamy *spójnym*, jeżeli każde dwa jego wierzchołki można połączyć drogą.

Definicja 2.33 (las)**.** *Lasem* nazywamy graf niezawierający cykli.

Definicja 2.34 (drzewo)**.** *Drzewem* nazywamy spójny las.

Definicja 2.35 (drzewo skierowane)**.** *Drzewem skierowanym* nazywamy graf skierowany, w którym:

— jeden z wierzchołków nie ma poprzedników i istnieje z niego droga do każdego innego wierzchołka (wierzchołek ten nazywamy *korzeniem*),

— każdy wierzchołek poza korzeniem ma dokładnie jeden poprzednik.

Definicja 2.36 (ojciec, syn)**.** Jeżeli $\Gamma = (U, E)$ jest drzewem skierowanym, $v \in U$, to *ojcem* wierzchołka v nazywamy poprzednik tego wierzchołka, zaś *synem* wierzchołka v — jego następnik.

Definicja 2.37 (przodek, potomek)**.** Jeżeli w drzewie skierowanym istnieje droga z wierzchołka u do wierzchołka v, to u nazywamy *przodkiem* wierzchołka v, zaś v — *potomkiem* wierzchołka u.

W szczególności, każdy wierzchołek drzewa skierowanego jest swoim potomkiem i przodkiem.

Definicja 2.38 (liść)**.** *Liściem* nazywamy wierzchołek drzewa nieposiadający synów.

Definicja 2.39 (uporządkowane drzewo skierowane)**.** *Uporządkowanym drzewem skierowanym* nazywamy drzewo skierowane, w którym synowie każdego wierzchołka są uporządkowani (od lewej do prawej).

W niniejszej pracy będę rozważał wyłącznie uporządkowane drzewa skierowane, dlatego będę nazywał je krótko *drzewami*.

2.4. Podstawowe pojęcia i definicje z zakresu teorii języków formalnych[3]

2.4.1. Symbole, alfabety, łańcuchy, języki

Pojęcie 2.40 (symbol)**.** Dowolny pojedynczy znak — literę, cyfrę, wyraz — będziemy nazywać *symbolem*. Pojęcie symbolu będziemy traktować jako pojęcie pierwotne, nieposiadające ścisłej definicji.

Definicja 2.41 (alfabet)**.** Dowolny skończony zbiór symboli nazywamy *alfabetem*.

Uwaga 2.1 (słownik)**.** Czasami, zwłaszcza gdy rozważanymi symbolami będą wyrazy języka naturalnego, będziemy mówić *słownik* zamiast *alfabet*. Takie nazewnictwo jest wówczas bardziej intuicyjne i zapobiega nieporozumieniom.

Przykład 2.1 (alfabety)**.** Za alfabety możemy uważać następujące zbiory:

(a) $\{A, B, C, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y, Z\}$,
(b) $\{a, ą, b, c, ć, d, e, ę, f, g, h, i, j, k, l, ł, m, n, ń, o, ó, p, r, s, ś, t, u, w, y, z, ź, ż\}$,
(c) $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$,
(d) $\{(0), (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), \ldots, (59)\}$,
(e) $\{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, o, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$,
(f) $\{\aleph, \imath, \sharp, \exists, \partial, \heartsuit\}$,
(g) { *gawron* , *gołąb* , *kos* , *szpak* , *wróbel* , *ćwierka* , *kracze* , *śpiewa* } .

[3] Na podstawie [2] i [5]

Elementy tych zbiorów to symbole. Zbiór z ostatniego przykładu (g) wygodniej byłoby nazwać raczej słownikiem niż alfabetem.

Zbiór $\{0, 1, 2, 3, \ldots\}$ (zbiór wszystkich liczb naturalnych) nie jest alfabetem, ponieważ nie jest skończony.

Definicja 2.42 (słowo, łańcuch)**.** *Słowem* lub *łańcuchem nad alfabetem* V nazywamy skończony ciąg symboli tego alfabetu. W przypadku, gdy wiadomo, o jaki alfabet chodzi, będziemy mówić krótko: *słowo* lub *łańcuch*.

Łańcuch (słowo) przedstawiany jest jako napis, w którym symbole pisane są jeden za drugim.

Podobnie jak w przypadku *alfabetu* i *słownika*, również w tym przypadku występują dwa różne określenia na to samo pojęcie. Ponieważ w tej pracy będą występować przykłady odwołujące się do języka naturalnego, będę unikał stosowania określenia *słowo*, aby nie doprowadzać do nieporozumień. Określenie *łańcuch* jest dużo bardziej jednoznaczne.

Przykład 2.2 (łańcuchy)**.** Weźmy alfabet $V = \{a, b, c, d, e\}$. Wówczas łańcuchami nad alfabetem V są na przykład ciągi *abc*, *aaaab* czy *ebeced*.

Definicja 2.43 (długość łańcucha)**.** Liczbę symboli w łańcuchu w nazywamy *długością* łańcucha w i oznaczamy przez $|w|$.

Przykład 2.3 (długość łańcucha)**.** Łańcuch *aabcc* (nad alfabetem $\{a, b, c\}$) ma długość 5, a łańcuch *wróbel ćwierka* (nad słownikiem g z przykładu 2.1) ma długość 2. Możemy zatem napisać:

$$|aabcc| = 5,$$
$$|\, wróbel\ ćwierka\,| = 2.$$

Definicja 2.44 (łańcuch pusty)**.** Ciąg niezawierający żadnych symboli nazywamy *łańcuchem pustym* i oznaczamy symbolem ϵ.

Łańcuch pusty ma długość 0 ($|\epsilon| = 0$) i jest łańcuchem nad każdym alfabetem.

Definicja 2.45 (konkatenacja, złożenie)**.** Jeżeli wypiszemy kolejno wszystkie symbole jednego łańcucha, a zaraz za nimi kolejno wszystkie symbole drugiego łańcucha, to otrzymany w ten sposób nowy łańcuch nazywamy *konkatenacją* albo *złożeniem* tych dwóch łańcuchów. Podobnie definiujemy konkatenację większej liczby łańcuchów.

Przykład 2.4 (konkatenacja)**.** Konkatenacją łańcuchów *abc* i $defg$ jest łańcuch $abcdefg$.

Konkatenacją łańcuchów *pq*, *r*, *stu* i *vwxyx* jest łańcuch *pqrstuvwxyz*.

Konkatenację n łańcuchów oznaczonych przez w_1, w_2, ..., w_n oznaczamy przez zestawienie obok siebie oznaczeń tych łańcuchów, w tym przypadku przez $w_1 w_2 \dots w_n$.

Bezpośrednio z definicji operacji konkatenacji wynika jej łączność:

$$x(yz) = xyz = (xy)z.$$

Definicja 2.46 (podłańcuch)**.** Łańcuch w' jest podłańcuchem łańcucha w, jeżeli istnieją łańcuchy u, v takie, że $w = uw'v$.

Łańcuch pusty jest podłańcuchem każdego łańcucha.

Przykład 2.5 (podłańcuchy)**.** Podłańcuchami łańcucha *abcde* są na przykład łańcuchy ϵ, *a*, *abc*, *bcd*, *de*, *abcde*.

Definicja 2.47 (język)**.** Dowolny podzbiór zbioru wszystkich słów nad danym alfabetem nazywamy *językiem*.

Zbiór wszystkich słów nad danym alfabetem V jest również językiem i oznacza się go symbolem V^*.

Język wszystkich słów nad danym alfabetem V z wyjątkiem słowa pustego ϵ oznacza się symbolem V^+.

Przez V^n będziemy rozumieć zbiór $\{w \in V^* \colon |w| = n\}$.

Przykład 2.6 (języki)**.** Niech dany będzie alfabet $V = \{a, b, c\}$. Przykładami języków złożonych z łańcuchów nad tym alfabetem są między innymi następujące zbiory łańcuchów:

(a) $V^* = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \dots, ccc, aaaa, \dots, cccc, \dots\}$,
(b) $V^+ = \{a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \dots, ccc, aaaa, \dots, cccc, \dots\}$,
(c) $\{a, aa, aaa, aaaa, \dots\}$,
(d) $\{\epsilon, abc, cab, cba\}$,
(e) $\{\epsilon\}$,
(f) $\emptyset$.

Zauważmy, że języki utworzone z łańcuchów nad (skończonym) alfabetem mogą być zarówno zbiorami skończonymi, jak i nieskończonymi, a nawet zbiorem pustym.

2.4.2. Automaty

Definicja 2.48 (deterministyczny automat skończony)**.** *Deterministycznym automatem skończonym* nazywamy strukturę $M = (Q, \Sigma, \delta, q_0, F)$, gdzie:

— Q jest skończonym zbiorem, nazywanym *zbiorem stanów*,
— Σ jest skończonym alfabetem, nazywanym *alfabetem wejściowym*,

— funkcja $\delta\colon Q \times \Sigma \to Q$ jest nazywana *funkcją przejścia*,
— wyróżniony stan $q_0 \in Q$ nazywany jest *stanem początkowym*,
— zbiór $F \subseteq Q$ nazywany jest *zbiorem stanów końcowych*.

Definicja 2.49. Dla deterministycznego automatu skończonego $M = (Q, \Sigma, \delta, q_0, F)$ funkcję $\hat{\delta}\colon Q \times \Sigma^* \to Q$ definiujemy następująco:

$$\begin{aligned} \hat{\delta}(q, \epsilon) &:= q, \\ \hat{\delta}(q, wa) &:= \delta(\hat{\delta}(q, w), a) \qquad \text{dla dowolnych } w \in \Sigma^*, a \in \Sigma. \end{aligned}$$

Innymi słowy, wartość $\hat{\delta}(q, w)$ jest stanem, w jakim znajdzie się automat M po odczytaniu łańcucha w.

Definicja 2.50 (język akceptowany przez deterministyczny automat skończony)**.** *Językiem akceptowanym przez deterministyczny automat skończony* $M = (Q, \Sigma, \delta, q_0, F)$ nazywamy język

$$L(M) := \{w \in \Sigma^*\colon \hat{\delta}(q_0, w) \in F\}.$$

O dowolnym łańcuchu $w \in L(M)$ mówimy, że jest *łańcuchem akceptowanym przez automat* M.

Definicja 2.51 (język regularny)**.** *Językiem regularnym* nazywamy każdy język, który jest akceptowany przez jakiś deterministyczny automat skończony.

Definicja 2.52 (niedeterministyczny automat skończony)**.** *Niedeterministycznym automatem skończonym* nazywamy strukturę $M = (Q, \Sigma, \delta, q_0, F)$, gdzie:
— Q jest skończonym zbiorem, nazywanym *zbiorem stanów*,
— Σ jest skończonym alfabetem, nazywanym *alfabetem wejściowym*,
— funkcja $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$ jest nazywana *funkcją przejścia*,
— wyróżniony stan $q_0 \in Q$ nazywany jest *stanem początkowym*,
— zbiór $F \subseteq Q$ nazywany jest *zbiorem stanów końcowych*.

Definicja 2.53. Dla niedeterministycznego automatu skończonego $M = (Q, \Sigma, \delta, q_0, F)$ funkcję $\hat{\delta}\colon Q \times \Sigma^* \to \mathcal{P}(Q)$ definiujemy następująco:

$$\begin{aligned} \hat{\delta}(q, \epsilon) &:= \{q\}, \\ \hat{\delta}(q, wa) &:= \bigcup_{r \in \hat{\delta}(q,w)} \delta(r, a) \qquad \text{dla dowolnych } w \in \Sigma^*, a \in \Sigma. \end{aligned}$$

Innymi słowy, wartość $\hat{\delta}(q, w)$ jest zbiorem możliwych stanów, w jakich może znaleźć się automat M po odczytaniu łańcucha w.

Definicja 2.54 (język akceptowany przez niedeterministyczny automat skończony)**.** *Językiem akceptowanym przez niedeterministyczny automat skończony* $M = (Q, \Sigma, \delta, q_0, F)$ nazywamy język

$$L(M) := \{w \in \Sigma^*\colon \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

O dowolnym łańcuchu $w \in L(M)$ mówimy, że jest *łańcuchem akceptowanym przez automat* M.

Twierdzenie 2.11 (równoważność deterministycznych i niedeterministycznych automatów skończonych)**.** *Dla dowolnego deterministycznego automatu skończonego M istnieje niedetermnistyczny automat skończony M' taki, że $L(M') = L(M)$. Podobnie, dla dowolnego niedeterministycznego automatu skończonego M istnieje determnistyczny automat skończony M' taki, że $L(M') = L(M)$.*

Dowód. Dowód tego twierdzenia można znaleźć w [2]. □

Wniosek 2.12. *Każdy niedeterministyczny automat skończony akceptuje język regularny. Każdy język regularny jest akceptowany przez jakiś niedeterministyczny automat skończony.*

Definicja 2.55 (niedeterministyczny automat skończony z ϵ-przejściami)**.** *Niedeterministycznym automatem skończonym z ϵ-przejściami* nazywamy strukturę $M = (Q, \Sigma, \delta, q_0, F)$, gdzie:

- Q jest skończonym zbiorem, nazywanym *zbiorem stanów*,
- Σ jest skończonym alfabetem, nazywanym *alfabetem wejściowym*,
- funkcja $\delta\colon Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$ jest nazywana *funkcją przejścia*,
- wyróżniony stan $q_0 \in Q$ nazywany jest *stanem początkowym*,
- zbiór $F \subseteq Q$ nazywany jest *zbiorem stanów końcowych*.

Powiemy, że stan $q' \in Q$ jest *osiągalny za pomocą samych ϵ-przejść* ze stanu $q \in Q$, jeżeli istnieje ciąg stanów $(q = q_0, q_1, \ldots, q_{n-1}, q_n = q')$ taki, że

$$\delta(q_0, \epsilon) = q_1, \quad \delta(q_1, \epsilon) = q_2, \quad \ldots, \quad \delta(q_{n-1}, \epsilon) = q_n.$$

Definicja 2.56 (ϵ-domknięcie)**.** Niech $M = (Q, \Sigma, \delta, q_0, F)$ będzie niedeterministycznym automatem skończonym z ϵ-przejściami. Wówczas zbiór wszystkich stanów osiągalnych za pomocą samych ϵ-przejść ze stanów ze zbioru $A \subseteq Q$ nazywamy *ϵ-domknięciem* zbioru A i oznaczamy przez $\mathrm{Domkn}_\epsilon(A)$.

Definicja 2.57. Dla niedeterministycznego automatu skończonego z ϵ-przejściami $M = (Q, \Sigma, \delta, q_0, F)$ funkcję $\hat{\delta}\colon Q \times \Sigma^* \to \mathcal{P}(Q)$ definiujemy w sposób rekurencyjny:

$$\hat{\delta}(q, \epsilon) := \mathrm{Domkn}_\epsilon(\{q\}),$$

$$\hat{\delta}(q, wa) := \mathrm{Domkn}_\epsilon\left(\bigcup_{r \in \hat{\delta}(q,w)} \delta(r, a)\right) \qquad \text{dla dowolnych } w \in \Sigma^*, a \in \Sigma.$$

Definicja 2.58 (język akceptowany przez niedeterministyczny automat skończony z ϵ-przejściami)**.** *Językiem akceptowanym przez niedeterministyczny automat skończony z ϵ-przejściami* $M = (Q, \Sigma, \delta, q_0, F)$ nazywamy język

$$L(M) := \{w \in \Sigma^* \colon \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

O dowolnym łańcuchu $w \in L(M)$ mówimy, że jest *łańcuchem akceptowanym przez automat* M.

Twierdzenie 2.13 (równoważność niedeterministycznych automatów skończonych z ϵ-przejściami i bez ϵ-przejść)**.** *Dla dowolnego niedeterministycznego automatu skończonego (bez ϵ-przejść) M istnieje niedetermnistyczny automat skończony z ϵ-przejściami M' taki, że $L(M') = L(M)$. Podobnie, dla dowolnego niedeterministycznego automatu skończonego z ϵ-przejściami M istnieje niedetermnistyczny automat skończony (bez ϵ-przejść) M' taki, że $L(M') = L(M)$.*

Dowód. Dowód tego twierdzenia można znaleźć w [2]. □

2.4.3. Gramatyki

Definicja 2.59 (gramatyka)**.** *Gramatyką (nieograniczoną)* nazywamy strukturę $G = (V, T, R, S)$, w której:

- V jest alfabetem, nazywanym *alfabetem symboli pomocniczych (nieterminalnych)* albo krótko *alfabetem zmiennych*,
- T jest rozłącznym z V alfabetem, nazywanym *alfabetem symboli końcowych (terminalnych)*,
- R jest zbiorem *reguł produkcji*, czyli napisów postaci $\zeta \to \xi$, gdzie $\zeta \in (V \cup T)^+$, $\xi \in (V \cup T)^*$,
- wyróżniony symbol $S \in V$ nazywany jest *symbolem początkowym.*

Jeżeli produkcja r jest postaci $\zeta \to \xi$, to łańcuch ζ będziemy nazywać *poprzednikiem produkcji* r, a łancuch ξ — *następnikiem produkcji* r.

Definicja 2.60 (gramatyka kontekstowa)**.** *Gramatyką kontekstową* nazywamy gramatykę $G = (V, T, R, S)$, której każda produkcja jest postaci $\zeta A \xi \to \zeta \omega \xi$, gdzie $A \in V$, $\zeta, \xi \in (V \cup T)^*$, $\omega \in (V \cup T)^+$.

Definicja 2.61 (gramatyka bezkontekstowa z ϵ-produkcjami)**.** *Gramatyką bezkontekstową z ϵ-produkcjami* nazywamy gramatykę $G = (V, T, R, S)$, której każda produkcja jest postaci $A \to \omega$, gdzie $A \in V$, $\omega \in (V \cup T)^*$.

Definicja 2.62 (gramatyka bezkontekstowa ϵ-wolna)**.** *Gramatyką bezkontekstową bez ϵ-produkcji* albo *gramatyką bezkontekstową ϵ-wolną* nazywamy gramatykę $G = (V, T, R, S)$, której każda produkcja jest postaci $A \to \omega$, gdzie $A \in V$, $\omega \in (V \cup T)^+$.

Twierdzenie 2.14. *Każda gramatyka bezkontekstowa ϵ-wolna jest gramatyką kontekstową.*

Dowód. Jeżeli w definicji gramatyki kontekstowej (definicja 2.60) przyjmiemy $\zeta = \xi = \epsilon$, otrzymamy definicję gramatyki bezkontekstowej ϵ-wolnej. □

Gramatyki bezkontekstowe ϵ-wolne będziemy krótko nazywać *gramatykami bezkontekstowymi*. W dalszej części pracy, ilekroć będziemy mówić o gramatykach bezkontekstowych z ϵ-produkcjami, będzie to zawsze wyraźnie zaznaczone.

Przykład 2.7 (gramatyka bezkontekstowa)**.** Przykład gramatyki bezkontekstowej $G = (V, T, R, S)$:
- $V = \{S\}$,
- $T = \{a\}$,
- $R = \{S \to SS,\ S \to a\}$,
- $S \in V$ jest symbolem początkowym.

Przykład 2.8 (gramatyka bezkontekstowa)**.** Inny przykład gramatyki bezkontekstowej $G = (V, T, R, S)$:
- $V = \{S, A, B\}$,
- $T = \{a, b\}$,
- $R = \{S \to AB,\ S \to BA,\ B \to AA,\ A \to a,\ A \to b\}$,
- $S \in V$ jest symbolem początkowym.

Przykład 2.9 (gramatyka bezkontekstowa)**.** Jeszcze inny przykład gramatyki bezkontekstowej $G = (V, T, R, S)$:
- $V = \{S\}$,
- $T = \{a, b, c\}$,
- $R = \{S \to aSb,\ S \to ab, S \to c\}$,
- $S \in V$ jest symbolem początkowym.

Definicja 2.63 (postać normalna Chomsky'ego)**.** Gramatyka bezkontekstowa $G = (V, T, R, S)$ jest w *postaci normalnej Chomsky'ego*, jeżeli każda jej produkcja ma postać $A \to BC$ lub $A \to a$, gdzie $A, B, C \in V$, $a \in T$.

Gramatyki z przykładów 2.7 i 2.8 są gramatykami w postaci normalnej Chomsky'ego, natomiast gramatyka z przykładu 2.9 już nie jest, ponieważ zawiera regułę $S \to aSb$, która ma więcej niż dwa symbole po prawej stronie, oraz regułę $S \to ab$, która ma po prawej stronie więcej niż jeden symbol końcowy.

Definicja 2.64 (gramatyka prawostronnie liniowa)**.** *Gramatyką prawostronnie liniową* nazywamy gramatykę formalną, której każda produkcja ma postać $A \to wB$ lub $A \to w$, gdzie $A, B \in V$, $w \in T^*$.

Definicja 2.65 (gramatyka lewostronnie liniowa)**.** *Gramatyką lewostronnie liniową* nazywamy gramatykę formalną, której każda produkcja ma postać $A \to Bw$ lub $A \to w$, gdzie $A, B \in V$, $w \in T^*$.

Definicja 2.66 (gramatyka regularna)**.** *Gramatyką regularną* nazywamy każdą gramatykę, która jest prawostronnie liniowa lub lewostronnie liniowa.

Twierdzenie 2.15. *Każda gramatyka regularna jest gramatyką bezkontekstową z ϵ-produkcjami.*

Dowód. $A \in V$. Jeżeli $B \in V$, $w \in T^*$, to $wB, w \in (V \cup T)^*$. □

Przykład 2.10 (gramatyka regularna)**.** Gramatyka z przykładu 2.7 nie jest regularna. Wystarczy jednak drobna modyfikacja, by uczynić z niej gramatykę prawostronnie liniową $G = (V, T, R, S)$:

— $V = \{S\}$,
— $T = \{a\}$,
— $R = \{S \to aS,\ S \to a\}$,
— $S \in V$ jest symbolem początkowym.

Definicja 2.67 (bezpośrednia wyprowadzalność)**.** Niech $G = (V, T, R, S)$ będzie dowolną gramatyką oraz $\zeta, \xi, \omega, \omega' \in (V \cup T)^*$. Jeżeli istnieje produkcja $r = (\omega \to \omega') \in R$, to łańcuch $\zeta\omega'\xi$ nazywamy *bezpośrednio wyprowadzalnym* z łańcucha $\zeta\omega\xi$ w gramatyce G (przy użyciu produkcji r). Zapisujemy ten fakt jako

$$\zeta\omega\xi \Rightarrow_G \zeta\omega'\xi.$$

Jeżeli jasne jest, o którą gramatykę chodzi, symbol gramatyki możemy pominąć:

$$\zeta\omega\xi \Rightarrow \zeta\omega'\xi.$$

Definicja 2.68 (wyprowadzenie)**.** Niech $G = (V, T, R, S)$ będzie gramatyką oraz $\omega, \omega' \in (V \cup T)^*$. *Wyprowadzeniem* łańcucha ω' z łańcucha ω w gramatyce G nazywamy wówczas ciąg łańcuchów $\zeta_0, \zeta_1, \ldots, \zeta_m \in (V \cup T)^*$, $m \geqslant 1$, taki, że:

$$\zeta_0 = \omega,\ \zeta_m = \omega',$$
$$\zeta_0 \Rightarrow_G \zeta_1,\ \zeta_1 \Rightarrow_G \zeta_2,\ \ldots,\ \zeta_{m-1} \Rightarrow_G \zeta_m.$$

Liczbę m nazywamy *długością wyprowadzenia*.

Definicja 2.69 (wyprowadzalność)**.** Łańcuch $\omega' \in (V \cup T)^*$ nazywamy *wyprowadzalnym* z łańcucha $\omega \in (V \cup T)^*$ w gramatyce $G = (V, T, R, S)$, jeżeli istnieje wyprowadzenie łańcucha ω' z łańcucha ω w gramatyce G. Wyprowadzalność łańcucha ω' z ω zapisujemy następująco:

$$\omega \Rightarrow_G^* \omega'.$$

Jeżeli nie prowadzi to do niejasności, można pominąć symbol gramatyki i napisać po prostu

$$\omega \Rightarrow^* \omega'.$$

Relacja wyprowadzalności w gramatyce $G = (V, T, R, S)$ jest zwrotna, tj. dla każdego łańcucha $\omega \in (V \cup T)^*$ zachodzi związek

$$\omega \Rightarrow_G^* \omega.$$

Przykład 2.11 (wyprowadzenie)**.** Niech dana będzie gramatyka G z przykładu 2.8. W gramatyce tej łańcuch ab jest wyprowadzalny z symbolu B, co możemy zapisać jako

$$B \Rightarrow^* ab.$$

Można się przekonać, że istotnie, wszystkie warunki ku temu są spełnione:

$$B \overset{①}{\Rightarrow} AA \overset{②}{\Rightarrow} aA \overset{③}{\Rightarrow} ab,$$

$$\begin{aligned} ① &\quad B \to AA \in R, \\ ② &\quad A \to a \in R, \\ ③ &\quad A \to b \in R. \end{aligned}$$

W dalszej części pracy, jeżeli będzie mowa o wyprowadzeniu pewnego łańcucha bez podania, z jakiego łańcucha został on wyprowadzony, będziemy przyjmować, że chodzi o wyprowadzenie z symbolu początkowego gramatyki.

Przykład 2.12 (wyprowadzenie z symbolu początkowego)**.** Niech dana będzie gramatyka G z przykładu 2.8. W gramatyce tej łańcuch aba jest wyprowadzalny z symbolu początkowego S gramatyki G:

$$S \Rightarrow^* aba.$$

Istotnie, wyprowadzeniem łańcucha aba jest ciąg

$$S \Rightarrow AB \Rightarrow AAA \Rightarrow AAa \Rightarrow Aba \Rightarrow aba.$$

Jeden łańcuch może posiadać kilka wyprowadzeń z danego symbolu (w tym również symbolu początkowego gramatyki):

Przykład 2.13 (różne wyprowadzenia tego samego łańcucha)**.** Niech dane będą gramatyka bezkontekstowa z przykładu 2.8 i łańcuch aba. Łańcuch ten jest wyprowadzalny z symbolu początkowego S, o czym przekonaliśmy się w przykładzie 2.12, konstruując odpowiednie wyprowadzenie. Z drugiej strony, ciąg

$$S \Rightarrow BA \Rightarrow AAA \Rightarrow AAa \Rightarrow Aba \Rightarrow aba.$$

także jest wyprowadzeniem łańcucha *aba* z symbolu S. Widać zatem, że istotnie, jeden łańcuch może posiadać więcej niż jedno wyprowadzenie z danego symbolu.

Definicja 2.70 (forma zdaniowa)**.** Niech $G = (V, T, R, S)$ będzie gramatyką. Łańcuch $\omega \in (V \cup T)^*$ nazywamy *formą zdaniową* tej gramatyki, jeżeli

$$S \Rightarrow_G^* \omega.$$

Definicja 2.71 (język generowany przez gramatykę)**.** *Język generowany przez gramatykę* $G = (V, T, R, S)$ definiujemy jako zbiór form zdaniowych tej gramatyki złożonych z samych symboli końcowych i oznaczamy przez $L(G)$:

$$L(G) := \{w \in T^* \colon S \Rightarrow_G^* w\}.$$

Definicja 2.72 (język kontekstowy)**.** *Językiem kontekstowym* nazywamy język generowany przez pewną gramatykę kontekstową.

Definicja 2.73 (język bezkontekstowy)**.** *Językiem bezkontekstowym* nazywamy język generowany przez pewną gramatykę bezkontekstową.

Uwaga 2.2. Ponieważ żadna gramatyka bezkontekstowa (ϵ-wolna) nie generuje łańcucha pustego ϵ, zatem w świetle powyższej defnicji żaden język bezkontekstowy nie zawiera łańcucha pustego ϵ.

Twierdzenie 2.16. *Każdy język bezkontekstowy jest językiem kontekstowym.*

Dowód. Teza wynika bezpośrednio z twierdzenia 2.14. □

Definicja 2.74 (gramatyki równoważne)**.** Gramatyki G_1 i G_2 nazywamy *równoważnymi*, jeżeli

$$L(G_1) = L(G_2).$$

Twierdzenie 2.17. *Jeżeli G jest gramatyką bezkontekstową z ϵ-produkcjami, to $L(G) \setminus \{\epsilon\}$ jest językiem bezkontekstowym.*

Dowód. W [2] dowiedzione jest twierdzenie, że jeśli G jest gramatyką bezkontekstową z ϵ-produkcjami, to istnieje gramatyka bezkontekstowa ϵ-wolna, która generuje język $L(G) \setminus \{\epsilon\}$. □

Twierdzenie 2.18. *Każdy język bezkontekstowy jest generowany przez pewną gramatykę bezkontekstową w postaci normalnej Chomsky'ego.*

Dowód. Dowód powyższego faktu polega na odpowiednim przekształceniu zestawu reguł gramatyki bezkontekstowej, tak aby reguły niebędące w postaci normalnej Chomsky'ego zostały zastąpione przez odpowiednie reguły postaci $A \to BC$ lub $A \to a$. Algorytm, za pomocą którego można to uzyskać, został przedstawiony m.in. w [2]. □

Twierdzenie 2.19. *Następujące warunki są równoważne:*

(1) Język L jest regularny.
(2) Istnieje gramatyka prawostronnie liniowa, która generuje język L.
(3) Istnieje gramatyka lewostronnie liniowa, która generuje język L.

Dowód. Dowody równoważności warunków składających się na twierdzenie można znaleźć w [2]. □

Twierdzenie 2.20. *Każdy język regularny, który nie zawiera łańcucha pustego ϵ, jest językiem bezkontekstowym.*

Dowód. Niech L będzie językiem regularnym, który nie zawiera łańcucha pustego ϵ (czyli $L \setminus \{\epsilon\} = L$). Na mocy twierdzenia 2.19 istnieje gramatyka regularna G, która generuje język L. G jest gramatyką bezkontekstową z ϵ-regułami (z twierdzenia 2.15). Z twierdzenia 2.17 wynika natomiast, że $L = L \setminus \{\epsilon\}$ jest językiem bezkontekstowym. □

Przykład 2.14 (język generowany przez gramatykę bezkontekstową)**.** Gramatyka bezkontekstowa G z przykładu 2.7 generuje język

$$L(G) = \{a, aa, aaa, \ldots\}.$$

Język $L(G)$ jest językiem bezkontekstowym.

Przykład 2.15 (język generowany przez gramatykę regularną)**.** Gramatyka regularna G z przykładu 2.10 generuje język

$$L(G) = \{a, aa, aaa, \ldots\}.$$

Jest to ten sam język, co w przykładzie 2.14. Widać, że gramatyki z przykładów 2.7 i 2.10 generują te same języki. Język $L(G)$ jest nie tylko językiem bezkontekstowym, ale też językiem regularnym.

Definicja 2.75 (drzewo wyprowadzenia)**.** Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Drzewo Γ jest *drzewem wyprowadzenia* (albo *drzewem rozkładu*) dla gramatyki G, jeżeli:

— każdy wierzchołek drzewa Γ etykietowany jest symbolem ze zbioru $V \cup T$,
— korzeń drzewa Γ etykietowany jest symbolem początkowym S gramatyki G,
— jeśli wierzchołek wewnętrzny drzewa Γ etykietowany jest symbolem A, to A jest symbolem nieterminalnym ($A \in V$),
— jeśli synowie wierzchołka drzewa Γ etykietowanego symbolem A mają kolejno etykiety $X_1, X_2, \ldots, X_k \in V \cup T$, to $A \to X_1 X_2 \ldots X_k$ jest produkcją ze zbioru R.

Ponieważ w drzewie synowie każdego wierzchołka są uporządkowani, można uporządkować wszystkie liście każdego drzewa wyprowadzenia w następujący sposób:

— Jeżeli liście $u, v \in U$ drzewa $\Gamma = (U, E)$ są synami tego samego wierzchołka x, to ich uporządkowanie pokrywa się z uporządkowaniem ich jako synów wierzchołka x,

— Jeżeli liście $u, v \in U$ drzewa $\Gamma = (U, E)$ mają różnych ojców, to istnieje dokładnie jeden taki wierzchołek x, że jest potomkiem każdego wspólnego przodka wierzchołków u i v. Wówczas dokładnie jeden z synów wierzchołka x jest przodkiem wierzchołka u (nazwijmy go u') i dokładnie jeden z synów wierzchołka x jest przodkiem wierzchołka v (nazwijmy go v'). Wtedy wierzchołki u, v jako liście drzewa Γ porządkujemy tak, jak uporządkowani są ich przodkowie u', v' jako synowie wierzchołka x.

Definicja 2.76 (plon)**.** Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. *Plonem* drzewa wyprowadzenia Γ nazywamy łańcuch $\omega \in (V \cup T)^+$ złożony z etykiet liści tego drzewa uporządkowanych w podany powyżej sposób.

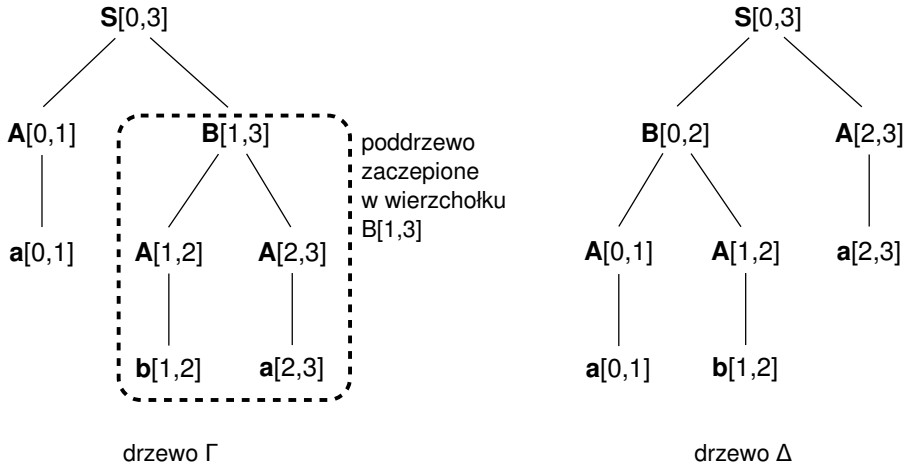
Definicja 2.77 (poddrzewo)**.** *Poddrzewem* drzewa wyprowadzenia Γ zaczepionym w wierzchołku x o etykiecie A nazywamy drzewo złożone ze wszystkich potomków wierzchołka x wraz z łączącymi je łukami.

Różne wierzchołki danego drzewa wyprowadzenia mogą być etykietowane tymi samymi symbolami gramatyki. Aby odróżnić różne wierzchołki o tych samych etykietach, wprowadzamy następującą umowę. Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Niech Γ będzie drzewem wyprowadzenia łańcucha $w = w_1 w_2 \dots w_n \in T^+$ w gramatyce G. Jeżeli plonem poddrzewa zaczepionego w wierzchołku x o etykiecie $X \in V \cup T$ jest łańcuch $w_{i+1} \dots w_j \in T^+$, to wierzchołek x będziemy oznaczać przez $X[i, j]$. Parę liczb (i, j) będziemy nazywać *zakresem* wierzchołka x.

Przykład 2.16 (drzewo wyprowadzenia)**.** Niech dana będzie gramatyka G z przykładu 2.8. Dwa przykładowe drzewa wyprowadzenia łańcucha *aba* pokazane są na rysunku 2.1. Wierzchołki oznaczono za pomocą etykiet i zakresów. Wyróżniono poddrzewo drzewa Γ zaczepione w wierzchołku $B[1, 3]$.

Twierdzenie 2.21. *Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Wówczas łańcuch $\omega \in (V \cup T)^+$ jest wyprowadzalny z symbolu początkowego gramatyki G wtedy i tylko wtedy, gdy ω jest plonem pewnego drzewa wyprowadzenia w gramatyce G.*

Dowód. Dowód powyższego twierdzenia jest przedstawiony w [2]. □



Rysunek 2.1. Przykładowe drzewa wyprowadzenia łańcucha *aba*. Wyróżniono poddrzewo drzewa Γ zaczepione w wierzchołku $B[1,3]$

Definicja 2.78 (wyprowadzenie lewostronne)**.** Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. *Wyprowadzeniem lewostronnym* nazywamy takie wyprowadzenie $\zeta_0, \zeta_1, \ldots, \zeta_m \in (V \cup T)^+$, w którym w każdym kroku dokonujemy zastąpienia zmiennej leżącej najbardziej na lewo. Innymi słowy, dla każdego $i \in \{1, \ldots, m\}$ zachodzi:

$$\zeta_{i-1} = uA\xi \Rightarrow u\omega\xi = \zeta_i$$

dla pewnych $u \in T^*$, $A \in V$, $\xi \in (V \cup T)^*$, $\omega \in (V \cup T)^+$ oraz dla $A \to \omega \in R$.

Wyprowadzenie lewostronne łańcucha $\omega \in (V \cup T)^+$ z symbolu początkowego S gramatyki bezkontekstowej $G = (V, T, R, S)$ będziemy nazywać krótko *wyprowadzeniem lewostronnym łańcucha* ω. Ilekroć będzie mowa o wyprowadzeniu lewostronnym łańcucha ω z symbolu (bądź łańcucha) innego niż symbol początkowy, będzie to wyraźnie zaznaczone.

Jeden łańcuch może posiadać kilka różnych wyprowadzeń lewostronnych.

Przykład 2.17. Niech dana będzie gramatyka G z przykładu 2.8. Wyprowadzeniem lewostronnym łańcucha *aba* w gramatyce G jest ciąg

$$S \Rightarrow AB \Rightarrow aB \Rightarrow aAA \Rightarrow abA \Rightarrow aba.$$

Innym wyprowadzeniem lewostronnym tego samego łańcucha jest ciąg

$$S \Rightarrow BA \Rightarrow AAA \Rightarrow aAA \Rightarrow abA \Rightarrow aba.$$

Widać więc, że istotnie jeden łańcuch może mieć więcej niż jedno wyprowadzenie lewostronne.

Twierdzenie 2.22. *Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową oraz $u \in T^+$. Wówczas następujące warunki są równoważne:*

(1) $S \Rightarrow^* u$.
(2) Istnieje drzewo wyprowadzenia w gramatyce G o plonie u.
(3) Istnieje wyprowadzenie lewostronne łańcucha u w gramatyce G.

Dowód. Dowód można znaleźć w [2]. □

Jeżeli w gramatyce bezkontekstowej $G = (V, T, R, S)$ łańcuch $\omega \in (V \cup T)^+$ posiada wyprowadzenie lewostronne, to istnieje drzewo wyprowadzenia w gramatyce G o plonie ω.

Definicja 2.79 (podwyprowadzenie)**.** Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Niech l będzie wyprowadzeniem lewostronnym

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow \zeta_m,$$

zaś l' wyprowadzeniem lewostronnym

$$S = \xi_0 \Rightarrow \xi_1 \Rightarrow \ldots \Rightarrow \xi_n$$

Wyprowadzenie l' nazywamy *podwyprowadzeniem* wyprowadzenia l, jeżeli dla każdego $i = 0, 1, \ldots, n$ zachodzi $\xi_i = \zeta_i$.

Ponadto podwyprowadzenie l' wyprowadzenia l nazywamy *właściwym*, jeżeli $l' \neq l$.

Rozdział 3

Automaty probabilistyczne

3.1. Probabilistyczne automaty skończone[1]

Definicja 3.1 (probabilistyczny automat skończony)**.** *Probabilistycznym automatem skończonym* nazywamy strukturę $M = (Q, \Sigma, P, \pi, f)$, gdzie:

- Q jest skończonym zbiorem, nazywanym *zbiorem stanów*,
- Σ jest skończonym alfabetem, nazywanym *alfabetem wejściowym*,
- funkcja $P\colon Q \times \Sigma \times Q \to [0,1]$, nazywana *prawdopodobieństwem przejścia*, spełnia warunek:

$$\sum_{q' \in Q} P(q, a, q') \;=\; 1 \quad \text{dla dowolnych } q \in Q, a \in \Sigma,$$

- wektor wierszowy $\pi = [\pi_q]_{q \in Q} \in [0,1]^{|Q|}$ nazywany *początkowym rozkładem prawdopodobieństwa*, spełnia warunek:

$$\sum_{q \in Q} \pi_q \;=\; 1,$$

- wektor $f \in \{0,1\}^{|Q|}$ jest wektorem kolumnowym.

Definicja 3.2. Dla dowolnego $a \in \Sigma$ definiujemy macierz $P_a \in \mathbb{R}^{|Q| \times |Q|}$ jako

$$P_a = [P(q, a, q')]_{q, q' \in Q}\,.$$

Definicja 3.3. Dla dowolnego $w = w_1 \dots w_n \in \Sigma^+$ definiujemy macierz $P_w \in \mathbb{R}^{|Q| \times |Q|}$ jako

$$P_w = P_{w_1} P_{w_2} \cdots P_{w_n}.$$

P_ϵ definiujemy jako macierz identycznościową rzędu $|Q|$:

$$P_\epsilon := \mathbf{I}_{|Q|}.$$

Własność 3.1. *Macierze P_a i P_w z powyższych definicji są* wierszowo stochastyczne*, tj. suma wartości w każdym wierszu wynosi* 1*.*

[1] Na podstawie [4] i [8].

Dowód. Własność ta wynika z warunku na prawdopodobieństwo przejścia występującego w definicji 3.1 oraz z faktu, że iloczyn macierzy wierszowo stochastycznych jest macierzą wierszowo stochastyczną. □

Definicja 3.4 (prawdopodobieństwo łańcucha)**.** Niech dany będzie probabilistyczny automat skończony $M = (Q, \Sigma, P, \pi, f)$. Wówczas *prawdopodobieństwem akceptacji łańcucha* $w \in \Sigma^*$ *przez automat* M lub krótko *prawdopodobieństwem łańcucha* w nazywamy liczbę

$$\mathbf{P}_M(w) := \pi P_w f.$$

Definicja 3.5 (język akceptowany przez probabilistyczny automat skończony)**.** *Językiem akceptowanym przez probabilistyczny automat skończony* $M = (Q, \Sigma, P, \pi, f)$ *z prawdopodobieństwem większym niż* η nazywamy język

$$L_\eta(M) := \{w \in \Sigma^* \colon \mathbf{P}_M(w) > \eta\} = \{w \in \Sigma^* \colon \pi P_w f > \eta\},$$

czyli język łańcuchów, których prawdopodobieństwo akceptacji przez automat M wynosi więcej niż η.

Definicja 3.6 (język η-stochastyczny)**.** Język L nazywamy *η-stochastycznym* (dla $0 \leqslant \eta < 1$), jeżeli istnieje probabilistyczny automat skończony M taki, że $L = L_\eta(M)$.

Definicja 3.7 (język stochastyczny)**.** Język L nazywamy *stochastycznym*, jeżeli istnieje $0 \leqslant \eta < 1$ takie, że L jest językiem η-stochastycznym.

Twierdzenie 3.2. *Każdy język regularny jest stochastyczny.*

Dowód. Niech L będzie językiem regularnym. Istnieje niedeterministyczny automat skończony $M = (Q, \Sigma, \delta, q_0, F)$ taki, że $L = L(M)$. Zdefiniujmy funkcję $P \colon Q \times \Sigma \times Q \to [0, 1]$ w następujący sposób:

$$P(q, a, q') := \begin{cases} \frac{1}{|\delta(q,a)|} & \text{dla } q' \in \delta(q, a), \\ 0 & \text{dla } q' \notin \delta(q, a). \end{cases}$$

Niech $\pi = [\pi_q]_{q \in Q}$ będzie wektorem wierszowym, którego elementy są określone następująco:

$$\pi_q := \begin{cases} 1 & \text{dla } q = q_0, \\ 0 & \text{dla } q \neq q_0. \end{cases}$$

Niech $f = [f_q]_{q \in Q}$ będzie wektorem kolumnowym, którego elementy są określone następująco:

$$f_q := \begin{cases} 1 & \text{dla } q \in F, \\ 0 & \text{dla } q \notin F. \end{cases}$$

Wówczas $M' = (Q, \Sigma, P, \pi, f)$ jest probabilistycznym automatem skończonym.

Udowodnimy teraz metodą indukcji względem długości łańcucha w następujący fakt: jeżeli $q \in \hat{\delta}(q_0, w)$, to

$$(\pi P_w)_q > 0,$$

czyli q-ty element wektora πP_w jest niezerowy.

Najpierw sprawdzamy, że teza zachodzi dla $|w| = 0$, czyli $w = \epsilon$. Mamy ciąg równoważnych formuł:

$$q \in \hat{\delta}(q_0, \epsilon),$$
$$q \in \{q_0\},$$
$$q = q_0,$$

a zatem

$$(\pi P_\epsilon)_q = \pi_q = \pi_{q_0} = 1 > 0.$$

Teraz przeprowadzamy krok indukcyjny. Załóżmy, że teza zachodzi dla dowolnego łańcucha w takiego, że $|w| = n$. Weźmy łańcuch w', $|w'| = n + 1$. Łańcuch w' możemy przedstawić jako $w' = wa$, gdzie $|w| = n$, $a \in \Sigma$. Niech $q \in \hat{\delta}(q_0, w')$. Wówczas:

$$q \in \hat{\delta}(q_0, wa),$$
$$q \in \bigcup_{r \in \hat{\delta}(q_0, w)} \delta(r, a),$$
$$q \in \delta(r_0, a) \quad \text{dla pewnego } r_0 \in \hat{\delta}(q_0, w).$$

Zachodzą równości:

$$(\pi P_{wa})_q = (\pi P_w P_a)_q = \sum_{r \in Q} (\pi P_w)_r (P_a)_{r,q} \geqslant (\pi P_w)_{r_0} (P_a)_{r_0,q}.$$

Ponieważ $r_0 \in \hat{\delta}(q_0, w)$, zatem na mocy założenia indukcyjnego

$$(\pi P_w)_{r_0} > 0.$$

Z kolei z definicji macierzy P_a mamy, że

$$(P_a)_{r_0,q} = P(r_0, a, q) = \frac{1}{|\delta(r_0, a)|} > 0,$$

ponieważ $q \in \delta(r_0, a)$. Stąd ostatecznie

$$(\pi P_{w'})_q > 0,$$

co kończy krok indukcyjny.

Teraz pokażemy, że automat probabilistyczny M' akceptuje język L z prawdopodobieństwem większym niż 0. Niech $w \in L = L(M)$, czyli

$$\hat{\delta}(q_0, w) \cap F \neq \emptyset.$$

Wynika stąd, że istnieje stan końcowy $q_\mathrm{f} \in F$ taki, że $q_\mathrm{f} \in \hat{\delta}(q_0, w)$. Z własności, której dowiedliśmy przed chwilą, wynika, że

$$(\pi P_w)_{q_\mathrm{f}} > 0.$$

Z drugiej strony, $q_\mathrm{f} \in F$ oznacza, że

$$f_{q_\mathrm{f}} > 0.$$

Mamy zatem:

$$\mathbf{P}_M(w) = (\pi P_w) f = \sum_{q \in Q} (\pi P_w)_q \, f_q \geqslant (\pi P_w)_{q_\mathrm{f}} \, f_{q_\mathrm{f}} > 0,$$

czyli

$$w \in L_0(M').$$

W ten sposób pokazaliśmy zawieranie

$$L(M) \subseteq L_0(M').$$

Podobnie pokazujemy inkluzję w drugą stronę:

$$L_0(M') \subseteq L(M).$$

Ostatecznie

$$L = L(M) = L_0(M').$$

Stąd L jest językiem 0-stochastycznym, a zatem jest też językiem stochastycznym. □

Twierdzenie 3.3. *Każdy język* 0*-stochastyczny jest językiem regularnym.*

Dowód. Niech L będzie językiem 0-stochastycznym. Istnieje probabilistyczny automat skończony $M = (Q, \Sigma, P, \pi, f)$ taki, że $L = L_0(M) = \{w \in \Sigma^* \colon \pi P_w f > 0\}$. Niech $Q' := Q \cup \{q_0\}$, gdzie $q_0 \notin Q$ jest nowo utworzonym dodatkowym stanem. Niech $F := \{q \in Q \colon f_q = 1\}$. Zdefiniujemy funkcję $\delta \colon Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$ w następujący sposób:

$$\delta(q, a) := \begin{cases} \emptyset & \text{dla } q \in Q, a = \epsilon, \\ \{q' \in Q \colon P(q, a, q') > 0\} & \text{dla } q \in Q, a \neq \epsilon, \\ \{q' \in Q \colon \pi_{q'} > 0\} & \text{dla } q = q_0, a = \epsilon, \\ \emptyset & \text{dla } q = q_0, a \neq \epsilon. \end{cases}$$

Wówczas $M' = (Q', \Sigma, \delta, q_0, F)$ jest niedeterministycznym automatem skończonym z ϵ-przejściami.

Udowodnimy teraz metodą indukcji względem długości łańcucha w następujący fakt: jeżeli $(\pi P_w)_q > 0$, to

$$q \in \hat{\delta}(q_0, w).$$

Najpierw sprawdzamy, że teza zachodzi dla $|w| = 0$, czyli $w = \epsilon$. Istotnie, następujące formuły wynikają jedna z drugiej:

$$\begin{gathered}(\pi P_\epsilon)_q > 0,\\ \pi_q > 0,\\ q \in \delta(q_0, \epsilon),\\ q \in \hat{\delta}(q_0, \epsilon).\end{gathered}$$

Teraz przeprowadzamy krok indukcyjny. Załóżmy, że teza zachodzi dla dowolnego łańcucha w takiego, że $|w| = n$. Weźmy łańcuch w', $|w'| = n + 1$. Łańcuch w' możemy przedstawić jako $w' = wa$, gdzie $|w| = n$, $a \in \Sigma$. Niech $(\pi P_{w'})_q > 0$. Wówczas:

$$\begin{gathered}(\pi P_{wa})_q > 0,\\ (\pi P_w P_a)_q > 0,\\ \sum_{r \in Q} (\pi P_w)_r (P_a)_{r,q} > 0,\\ (\pi P_w)_{r_0} > 0,\ (P_a)_{r_0,q} > 0 \quad \text{dla pewnego } r_0 \in Q.\end{gathered}$$

Na mocy założenia indukcyjnego $(\pi P_w)_{r_0} > 0$ oznacza, że

$$r_0 \in \hat{\delta}(q_0, w).$$

Z kolei $(P_a)_{r_0,q} > 0$ oznacza, że

$$P(r_0, a, q) > 0,$$

a zatem

$$q \in \delta(r_0, a).$$

Ostatecznie

$$q \in \bigcup_{r \in \hat{\delta}(q_0,w)} \delta(r,a) \subseteq \mathrm{Domkn}_\epsilon\left(\bigcup_{r \in \hat{\delta}(q_0,w)} \delta(r,a)\right) = \hat{\delta}(q_0, wa) = \hat{\delta}(q_0, w'),$$

co kończy krok indukcyjny.

Teraz pokażemy, że niedeterministyczny automat skończony z ϵ-przejściami M' akceptuje język L. Niech $w \in L = L_0(M)$, czyli

$$\pi P_w f > 0,$$
$$\sum_{q \in Q} (\pi P_w)_q \, f_q > 0.$$

Wynika stąd, że istnieje stan $q \in Q$ taki, że $f_q = 1$ oraz $(\pi P_w)_q > 0$. Oznacza to, że $q \in F$ oraz, na mocy dowiedzionego przed chwilą faktu, że $q \in \hat{\delta}(q_0, w)$. Stąd

$$F \cap \hat{\delta}(q_0, w) \neq \emptyset,$$

czyli

$$w \in L(M'),$$

zatem

$$L_0(M) \subseteq L(M').$$

W podobny sposób pokazujemy inkluzję w drugą stronę:

$$L(M') \subseteq L_0(M).$$

Ostatecznie otrzymujemy:

$$L = L_0(M) = L(M').$$

□

3.2. Łańcuchy Markowa[2]

Definicja 3.8 (łańcuch Markowa)**.** Niech $X_1, X_2, \ldots$ będzie (przeliczalnym) ciągiem dyskretnych zmiennych losowych, natomiast Q — skończonym lub przeliczalnym zbiorem *stanów*. Mówimy, że ciąg $(X_n)_{n=1}^{\infty}$ jest *łańcuchem Markowa*, jeżeli spełnia następującą *własność Markowa*:

$$\mathbf{P}(X_{n+1} = q | X_1 = q_1, \ldots, X_n = q_n) \;=\; \mathbf{P}(X_{n+1} = q | X_n = q_n)$$

(dla każdego $n \geqslant 1$, dla dowolnych stanów $q, q_1, q_2, \ldots, q_n \in Q$).

W dalszej części pracy będę rozważać jedynie łańcuchy Markowa, których zbiór stanów Q jest skończony.

[2]Na podstawie [4], [6] i [7].

Definicja 3.9 (jednorodny łańcuch Markowa)**.** Łańcuch Markowa $(X_n)_{n=1}^{\infty}$ nazywamy *jednorodnym*, jeżeli dla dowolnych $m, n = 1, 2, \ldots$ i dla dowolnych stanów $q, r \in Q$ zachodzi

$$\mathbf{P}(X_{m+1} = q | X_m = r) \;=\; \mathbf{P}(X_{n+1} = q | X_n = r).$$

Jeżeli łańcuch Markowa jest jednorodny, to prawdopodobieństwo przejścia z jednego stanu w drugi ($\mathbf{P}(X_{n+1} = q | X_n = r)$) nie zależy od czasu (n).

Definicja 3.10 (macierz przejścia)**.** Niech $(X_n)_{n=1}^{\infty}$ będzie jednorodnym łańcuchem Markowa ze zbiorem stanów $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$. *Macierzą przejścia* dla łańcucha $(X_n)_{n=1}^{\infty}$ nazywamy macierz A o wymiarach $|Q| \times |Q|$, której (i, j)-te elementy są postaci

$$A_{i,j} = \mathbf{P}(X_{n+1} = q_j | X_n = q_i).$$

Twierdzenie 3.4. *Macierz przejścia dla dowolnego łańcucha Markowa jest macierzą wierszowo stochastyczną.*

Dowód. Dla każdego $i = 1, 2, \ldots$ zachodzi:

$$\sum_{j=1}^{|Q|} p_{i,j} = \sum_{j=1}^{|Q|} \mathbf{P}(X_{n+1} = q_j | X_n = q_i) =$$
$$= \mathbf{P}\Big((X_{n+1} = q_1) \cup (X_{n+1} = q_2) \cup \ldots \cup (X_{n+1} = q_{|Q|}) \;\Big|\; X_n = q_i\Big) = 1.$$

□

Jednorodny łańcuch Markowa mażemy traktować jako trójkę uporządkowaną (Q, A, π), w której:
— Q jest skończonym zbiorem stanów,
— A jest wierszowo stochastyczną macierzą przejścia o wymiarach $|Q| \times |Q|$, której elementy są liczbami rzeczywistymi z przedziału $[0, 1]$,
— $\pi \in \mathbb{R}^{|Q|}$ jest wektorem wierszowym, którego elementy są liczbami z przedziału $[0, 1]$ i sumują się do 1.

Definicja 3.11 (model Markowa N-tego rzędu)**.** Niech $X_1, X_2, \ldots$ będzie ciągiem dyskretnych zmiennych losowych, zaś Q — skończonym zbiorem stanów. Mówimy, że ciąg $(X_n)_{n=1}^{\infty}$ jest *modelem Markowa N-tego rzędu* ($N = 0, 1, \ldots$), jeżeli spełnia następującą *własność Markowa N-tego rzędu*:

$$\mathbf{P}(X_{n+1} = q | X_1 = q_1, \ldots, X_n = q_n) \;=$$
$$= \mathbf{P}(X_{n+1} = q | X_{n-N+1} = q_{n-N+1}, \ldots, X_n = q_n) \quad (3.1)$$

(dla każdego $n \geqslant N$, dla dowolnych stanów $q, q_1, q_2, \ldots, q_n \in Q$).

W szczególności, każdy łańcuch Markowa jest modelem Markowa pierwszego rzędu.

W modelu rzędu 0 równanie (3.1) przybiera postać

$$\mathbf{P}(X_{n+1} = q | X_1 = q_1, \ldots, X_n = q_n) \;=\; \mathbf{P}(X_{n+1} = q).$$

Definicja 3.12 (jednorodny model Markowa N-tego rzędu)**.** Model Markowa N-tego rzędu $(X_n)_{n=1}^{\infty}$ nazywamy *jednorodnym*, jeżeli dla dowolnych $m, n = 1, 2, \ldots$ i dla dowolnych stanów $q, r_1, \ldots, r_N \in Q$ zachodzi równość:

$$\begin{aligned}\mathbf{P}(X_m = q | X_{m-N} = r_N, \ldots, X_{m-1} = r_1) \;=& \\ =\; \mathbf{P}(X_n = q | X_{n-N} = r_N, \ldots, X_{n-1} = r_1).&\end{aligned}$$

3.3. N-gramowe modele języka

3.3.1. Podstawowe definicje i własności[3]

Definicja 3.13 (model języka)**.** Niech V będzie alfabetem. *Modelem języka* nazywamy funkcję $P\colon V^* \to [0, 1]$ określoną na łańcuchach wyrazów, która dla każdego $n = 1, 2, \ldots$ spełnia warunek

$$\sum_{w \in V^n} P(w) \;=\; 1,$$

gdzie $V^n := \{w \in V^*\colon |w| = n\}$. Innymi słowy, model języka jest rozkładem prawdopodobieństwa na przestrzeni $\mathcal{P}(V^n)$ dla każdego n.

Wartość funkcji $P(w)$ nazywana jest często *prawdopodobieństwem łańcucha* w.

Definicja 3.14 (model N-gramowy)**.** Niech V będzie alfabetem. Niech ciąg zmiennych losowych $(X_n)_{n=1}^{\infty}$ będzie jednorodnym modelem Markowa $(N-1)$-szego rzędu (dla $N = 1, 2, \ldots$) ze zbiorem stanów V. Wówczas *modelem N-gramowym* nazywamy odwzorowanie $P\colon V^* \to [0, 1]$ określone następująco:

$$P(w_1 \ldots w_n) \;:=\; \prod_{i=1}^{n} \mathbf{P}(X_i = w_i | X_1 = w_1, \ldots, X_{i-1} = w_{i-1}). \tag{3.2}$$

Dla uproszczenia zapisu będę stosować notację

$$\mathbf{P}(w_i | w_1, \ldots, w_{i-1})$$

zamiast

$$\mathbf{P}(X_i = w_i | X_1 = w_1, \ldots, X_{i-1} = w_{i-1}).$$

[3]Na podstawie [4], [7], [9] i [11].

Zapis taki uzasadniony jest jednorodnością modelu Markowa.

W kontekście powyższego zapisu i własności modeli Markowa równanie (3.2) należy rozumieć jako:

$$\begin{aligned}
P(w_1 \dots w_n) &= \prod_{i=1}^{n} \mathbf{P}(w_i|w_1, \dots, w_{i-1}) = \\
&= \mathbf{P}(w_1)\,\mathbf{P}(w_2|w_1)\,\mathbf{P}(w_3|w_1, w_2) \cdots \mathbf{P}(w_n|w_1, \dots, w_{n-1}) = \\
&= \mathbf{P}(w_1)\,\mathbf{P}(w_2|w_1)\,\mathbf{P}(w_3|w_1, w_2) \cdots \mathbf{P}(w_N|w_1, \dots, w_{N-1}) \cdot \\
&\qquad \cdot \mathbf{P}(w_{N+1}|w_1, \dots, w_N) \cdots \mathbf{P}(w_n|w_1, \dots, w_{n-1}) = \\
&= \mathbf{P}(w_1)\,\mathbf{P}(w_2|w_1)\,\mathbf{P}(w_3|w_1, w_2) \cdots \mathbf{P}(w_N|w_1, \dots, w_{N-1}) \cdot \\
&\qquad \cdot \mathbf{P}(w_{N+1}|w_2, \dots, w_N) \cdots \mathbf{P}(w_n|w_{n-N+1}, \dots, w_{n-1}) = \\
&= \prod_{i=1}^{N} \mathbf{P}(w_i|w_1, \dots, w_{i-1}) \cdot \prod_{i=N+1}^{n} \mathbf{P}(w_i|w_{i-N+1}, \dots, w_{i-1})
\end{aligned}$$

Twierdzenie 3.5. *Model N-gramowy jest modelem języka.*

Dowód. Ustalmy dowolne $N \geqslant 0$. Aby udowodnić tezę twierdzenia, wystarczy pokazać, że dla dowolnego $n = 1, 2, \dots$ zachodzi równość:

$$\sum_{w_1 \dots w_n \in V^n} P(w_1 \dots w_n) = 1.$$

Dowód przeprowadzimy metodą indukcji matematycznej ze względu na n.

Dla $n = 1$ mamy:

$$\sum_{w_1 \in V} P(w_1) = \sum_{w_1 \in V} \mathbf{P}(X_1 = w_1) = 1.$$

Przyjmijmy teraz założenie indukcyjne, że zachodzi równość:

$$\sum_{w_1 \dots w_n \in V^n} P(w_1 \dots w_n) = 1.$$

Otrzymujemy wówczas:

$$\begin{aligned}
\sum_{w_1 \dots w_{n+1} \in V^{n+1}} P(w_1 \dots w_{n+1}) &= \sum_{w_1 \dots w_{n+1} \in V^{n+1}} \prod_{i=1}^{n+1} \mathbf{P}(w_i|w_1, \dots, w_{i-1}) = \\
&= \sum_{w_1 \dots w_{n+1} \in V^{n+1}} \left(\prod_{i=1}^{n} \mathbf{P}(w_i|w_1, \dots, w_{i-1}) \right) \mathbf{P}(w_{n+1}|w_1, \dots, w_n) = \\
&= \sum_{w_1 \dots w_n \in V^n} \sum_{w_{n+1} \in V} P(w_1 \dots w_n)\,\mathbf{P}(w_{n+1}|w_1, \dots, w_n) = \\
&= \sum_{w_1 \dots w_n \in V^n} \left(P(w_1 \dots w_n) \underbrace{\sum_{w_{n+1} \in V} \mathbf{P}(w_{n+1}|w_1, \dots, w_n)}_{=1} \right) = \\
&= \sum_{w_1 \dots w_n \in V^n} P(w_1 \dots w_n) = 1.
\end{aligned}$$

□

Na określenie kilku najpowszechniej stosowanych modeli N-gramowych używa się oddzielnych nazw. I tak, dla $N = 1$ mówimy o modelach *unigramowych*, dla $N = 2$ — o modelach *bigramowych*, zaś dla $N = 3$ — o modelach *trigramowych*.

N-gramowy model języka jest jednoznacznie określony przez wyznaczenie wszystkich wartości prawdopodobieństw warunkowych $\mathbf{P}(w_N|w_1, \ldots, w_{N-1})$ dla $w_1, \ldots, w_{N-1}, w_N \in V$. Model unigramowy jest wyznaczony przez wektor prawdopodobieństw $[p_i]_{i=1}^{|V|}$, $p_i := \mathbf{P}(w_i)$. Model bigramowy jest wyznaczony przez macierz prawdopodobieństw $[p_{i,j}]_{i=0,j=1}^{|V|,|V|}$, $p_{i,j} := \mathbf{P}(w_j|w_i)$, $p_{0,j} := \mathbf{P}(w_j)$.

3.3.2. Zastosowania modeli języka. Tworzenie modeli N-gramowych na podstawie danych statystycznych[4]

Celem modeli języka jest matematyczny opis pewnych statystycznych cech języków naturalnych, takich jak częstość występowania wyrazów i ciągów wyrazów w rzeczywistych tekstach. Modele języka znajdują zastosowanie w różnych narzędziach przetwarzania języka naturalnego: w rozpoznawaniu mowy, tłumaczeniu automatycznym czy wyszukiwaniu informacji. Wykorzystuje się je też w systemach ułatwiających komunikację osobom niepełnosprawnym. Stosując modele N-gramowe można nawet generować w automatyczny sposób teksty naśladujące język naturalny.

Aby zastosowanie modeli języka w takich aplikacjach było skuteczne, model powinien dobrze odzwierciedlać cechy statystyczne języka naturalnego. Aplikacja do automatycznego rozpoznawania mowy powinna znaleźć taki wyraz pasujący do nagrania, który z największym prawdopodobieństwem został wypowiedziany. Program do automatycznego poprawiania błędów korzysta z modelu języka, aby dopasować najbardziej prawdopodobną prawidłową formę błędnie napisanego wyrazu. Tłumacz automatyczny wykorzystuje model języka do znalezienia najbardziej prawdopodobnych słów języka docelowego, które odpowiadają tłumaczonemu pojęciu. Dlatego z punktu widzenia zastosowań modeli języka istotne jest, aby prawdopodobieństwa łańcuchów odpowiadały częstościom występowania poszczególnych łańcuchów w rzeczywistych tekstach.

Tekst, na podstawie którego będziemy tworzyć model języka, nazwiemy *korpusem*. Przez $C(w_1 \ldots w_n)$ będziemy oznaczać ilość wystąpień łańcucha wyrazów $w_1 \ldots w_n$ w korpusie.

Im większe N, tym dokładniejsze oszacowania prawdopodobieństw można uzyskać. Z drugiej jednak strony, wraz ze wzrostem N liczba parametrów

[4]Na podstawie [4] i [7].

potrzebnych do wyznaczenia rozkładu prawdopodobieństwa szybko rośnie. Dlatego w praktyce do modelowania języka najczęściej stosuje się modele bigramowe lub trigramowe.

Do najprostszych metod tworzenia N-gramowych modeli języka na podstawie danych statystycznych pochodzących z korpusów zaliczają się metoda największej wiarygodności oraz metody wygładzania, takie jak prawo Laplace'a, prawo Lidstone'a czy metoda Wittena-Bella.

Metoda największej wiarygodności[5] polega na tym, że przyjmujemy rozkład prawdopodobieństwa

$$\mathbf{P}(w_N|w_1,\ldots,w_{N-1}) := \frac{C(w_1\ldots w_{N-1}w_N)}{\sum_{a\in V} C(w_1\ldots w_{N-1}a)} = \tag{3.3}$$

$$= \frac{C(w_1\ldots w_{N-1}w_N)}{C(w_1\ldots w_{N-1})} \tag{3.4}$$

dla wszystkich $w_1,\ldots,w_{N-1},w_N \in V$. Jeżeli $C(w_1\ldots w_{N-1}) = 0$, to przyjmujemy $\mathbf{P}(w_N|w_1,\ldots,w_{N-1}) := 0$.

To podejście ma tę zaletę, że nie jest skomplikowane obliczeniowo, jednak posiada również pewne wady. Słowniki (V) języków naturalnych zawierają na ogół bardzo wiele wyrazów. Z tego powodu nawet w dużych korpusach zdecydowana większość możliwych łańcuchów N-wyrazowych nie będzie w ogóle występować, a zatem większość wartości $\mathbf{P}(w_N|w_1,\ldots,w_{N-1})$ będzie zerowa. Ponieważ wartości funkcji $P(w_1\ldots w_n)$ są określone jako iloczyny powyższych prawdopodobieństw warunkowych, funkcja P będzie przyjmowała wartość 0 dla bardzo wielu argumentów. Nie jest to zjawisko pożądane w zastosowaniach modeli języka. Korzystniej byłoby, gdyby dla takich łańcuchów funkcja P przyjmowała małe dodatnie wartości.

W tym celu stosuje się różne tzw. *metody wygładzania*. Jedną z metod wygładzania jest *prawo Laplace'a*, które polega na tym, że we wzorze (3.4) przyjmujemy o 1 większą liczność występowania łańcuchów.

$$\mathbf{P}(w_N|w_1,\ldots,w_{N-1}) := \frac{C(w_1\ldots w_{N-1}w_N)+1}{\sum_{a\in V}(C(w_1\ldots w_{N-1}a)+1)} =$$
$$= \frac{C(w_1\ldots w_{N-1}w_N)+1}{C(w_1\ldots w_{N-1})+|V|}$$

Uzyskane za pomocą prawa Laplace'a prawdopodobieństwa zależą od rozmiaru słownika $|V|$.

W metodzie Laplace'a prawdopodobieństwa przypisane łańcuchom niewystępującym w korpusie są na ogół przeszacowane. Można zaradzić temu

[5] Ang. *maximum likelihood estimation* [4] [7].

problemowi zwiększając liczności nie o 1, lecz o $\frac{1}{2}$ (*prawo Jeffreysa-Perksa*):

$$\mathbf{P}(w_N|w_1,\ldots,w_{N-1}) := \frac{C(w_1\ldots w_{N-1}w_N)+\frac{1}{2}}{\sum_{a\in V}\left(C(w_1\ldots w_{N-1}a)+\frac{1}{2}\right)} = \\ = \frac{C(w_1\ldots w_{N-1}w_N)+\frac{1}{2}}{C(w_1\ldots w_{N-1})+\frac{1}{2}|V|}$$

lub inną wartość pomiędzy 0 a 1 (*prawo Lidstone'a*):

$$\mathbf{P}(w_N|w_1,\ldots,w_{N-1}) := \frac{C(w_1\ldots w_{N-1}w_N)+\alpha}{\sum_{a\in V}\left(C(w_1\ldots w_{N-1}a)+\alpha\right)} = \\ = \frac{C(w_1\ldots w_{N-1}w_N)+\alpha}{C(w_1\ldots w_{N-1})+\alpha|V|}, \qquad 0\leqslant\alpha\leqslant 1.$$

Inną metodą wygładzania jest *metoda Wittena-Bella*, która opiera się na następującym pomyśle. Traktujemy każdy łańcuch N-wyrazowy o zerowej częstości w korpusie jako taki, który się *jeszcze* nie pojawił w korpusie — jeśliby się pojawił, to byłby to pierwszy raz, kiedy rejestrujemy taki łańcuch N-wyrazowy. Możemy zatem przybliżyć prawdopodobieństwo łańcucha N-wyrazowego o zerowej częstości w korpusie za pomocą prawdopodobieństwa zarejestorwania danego łańcucha N-wyrazowego po raz pierwszy. Metodę Wittena-Bella można streścić w zdaniu: *użyj liczby słów, które widziałeś raz, do oszacowania liczby słów, których jeszcze nie widziałeś* [4].

Prawdopodobieństwo zaobserwowania danego łańcucha N-wyrazowego po raz pierwszy szacujemy przez liczbę wystąpień łańcuchów N-wyrazowych każdego rodzaju po raz pierwszy. Ta liczba jest równa z kolei liczbie typów łańcuchów N-wyrazowych w korpusie.

Liczba łańcuchów o długości N w korpusie W wynosi $|W|-N+1\approx|W|$. Liczbę różnych rodzajów łańcuchów o długości N w korpusie W oznaczmy przez

$$t := \left|\{w\in V^N\colon w \text{ jest podciągiem } W\}\right|.$$

Zgodnie z założeniami metody Wittena-Bella rozkład prawdopodobieństwa $\mathbf{P}$ powinien być taki, aby spełniona była równość:

$$\sum_{C(w_1\ldots w_N)=0} P(w_1\ldots w_N) = \frac{t}{|W|+t}.$$

Proces wyznaczania prawdopodobieństw łańuchów zgodnie z tymi założeniami został opisany w [4] i [7].

Rozdział 4

Gramatyki probabilistyczne

4.1. Pojęcie probabilistycznych gramatyk bezkontekstowych

Definicja 4.1 (probabilistyczna gramatyka bezkontekstowa)**.** *Probabilistyczną (albo stochastyczną) gramatyką bezkontekstową* nazywamy strukturę $G = (V, T, R, S, P)$, w której:

— V jest alfabetem, nazywanym *alfabetem symboli pomocniczych (nieterminalnych)* albo krótko *alfabetem zmiennych*,

— T jest rozłącznym z V alfabetem, nazywanym *alfabetem symboli końcowych (terminalnych)*,

— R jest zbiorem *produkcji*, czyli napisów postaci $A \to \omega$, gdzie $A \in V$, $\omega \in (V \cup T)^+$,

— wyróżniony symbol $S \in V$ nazywany jest *symbolem początkowym*,

— $P\colon R \to [0, 1]$ jest funkcją (nazywaną *prawdopodobieństwem reguły* albo *prawdopodobieństwem produkcji*) taką, że dla każdego symbolu pomocniczego $A \in V$ spełniony jest warunek:

$$\sum_{A \to \omega \,\in\, R} P(A \to w) \;=\; 1. \tag{4.1}$$

Wzór (4.1) mówi o tym, że suma prawdopodobieństw reguł o jednakowym poprzedniku wynosi 1.

Probabilistyczna gramatyka bezkontekstowa powstaje z gramatyki bezkontekstowej przez dołączenie do każdej reguły wartości mającej reprezentować prawdopodobieństwo zastosowania tej reguły.

Jeśli piątka (V, T, R, S, P) jest probabilistyczną gramatyką bezkontekstową, to czwórka (V, T, R, S) jest gramatyką bezkontekstową. Z tego powodu wszystkie własności i definicje, które odnoszą się do (nieprobabilistycznych) gramatyk bezkontekstowych, mają również zastosowanie do probabilistycznych gramatyk bezkontekstowych.

Pojęcie gramatyk probabilistycznych wprowadzono, aby umożliwić modelowanie probabilistycznych własności języków (zwłaszcza języków naturalnych). W językach naturalnych niektóre łańcuchy (zdania) występują częściej niż inne, różne konstrukcje językowe są stosowane z różną częstością przez użytkowników języka. Potrzebne było zatem stworzenie takiego modelu, który umożliwiłby przypisanie każdemu łańcuchowi języka generowanego przez daną gramatykę bezkontekstową wartości prawdopodobieństwa, z jakim łańcuch ten wystąpiłby w „losowo wygenerowanym" tekście. Oczywiście w tym momencie pojawia się pytanie, jak należy rozumieć tę ideę „losowego generowania" tekstu i w jaki sposób przyporządkować prawdopodobieństwa regułom produkcji tak, aby stworzyć wiarygodny model języka.

Wyobraźmy sobie zatem, że mamy daną gramatykę bezkontekstową $G = (V, T, R, S)$ i stosujemy następującą procedurę. Zaczynamy od symbolu początkowego S gramatyki. Znajdujemy wszystkie reguły, których poprzednikiem jest symbol S. Spośród tych reguł wybieramy jedną, powiedzmy $S \to w$, gdzie w jest pewnym łańcuchem, i stosujemy do symbolu S — zastępujemy symbol S łańcuchem w, uzyskując nowy łańcuch. Bierzemy teraz dowolny z symboli nowego łańcucha i powtarzamy całą procedurę (znalezienie reguł, których jest poprzednikiem, wybór jednej z nich i zastosowanie), tak jak uczyniliśmy to poprzednio dla symbolu S.

W ten sposób, po skończonej liczbie takich kroków, możemy uzyskać łańcuch symboli terminalnych należący do języka $L(G)$.

W każdym kroku następuje wybór reguły. Niech A oznacza bieżący symbol, który rozważamy w tym kroku. Zbiór reguł, których poprzednikiem jest symbol A, oznaczmy przez

$$R_A := \{r \in R \colon r = (A \to \omega) \text{ dla pewnego } \omega \in (V \cup T)^+\} \subseteq R.$$

Zdefiniujmy funkcję $\mathbf{P}_A \colon \mathcal{P}(R_A) \to \mathbb{R}$ wzorem:

$$\mathbf{P}_A(B) := \sum_{r \in B} P(r) \quad \text{dla każdego } B \subseteq R_A.$$

Wówczas zachodzi

Twierdzenie 4.1. *Struktura* $(R_A, \mathcal{P}(R_A), \mathbf{P}_A)$ *jest przestrzenią probabilistyczną.*

Dowód. Teza wynika z faktu, że funkcja P jest rozkładem prawdopodobieństwa na zbiorze R_A, i z twierdzenia 2.3. □

Przykład 4.1 (probabilistyczna gramatyka bezkontekstowa)**.** Z gramatyki z przykładu 2.8 można uczynić probabilistyczną gramatykę bezkontekstową przyporządkowując jej regułom prawdopodobieństwa w odpowiedni sposób:

— $V = \{S, A, B\}$,
— $T = \{a, b\}$,
— $R = \{S \to AB,\ S \to BA,\ B \to AA,\ A \to a,\ A \to b\}$,
— $S \in V$ jest symbolem początkowym,
— wartości prawdopodobieństw reguł:
$P(S \to AB) = \frac{3}{4}$,
$P(S \to BA) = \frac{1}{4}$,
$P(B \to AA) = 1$,
$P(A \to a) = \frac{2}{3}$,
$P(A \to b) = \frac{1}{3}$,

4.2. Prawdopodobieństwa wyprowadzeń, drzew i łańcuchów

Definicja 4.2 (prawdopodobieństwo wyprowadzenia)**.** Niech dane będą:
— probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$,
— łańcuchy $\omega, \omega' \in (V \cup T)^+$,
— wyprowadzenie $l = (\zeta_0, \zeta_1, \ldots, \zeta_m)$ łańcucha ω' z łańcucha ω w gramatyce G:

$$\omega = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow \zeta_m = \omega',$$

takie, że łańcuch ζ_i wyprowadzono z łańcucha ζ_{i-1} przy użyciu produkcji r_i.

Wówczas *prawdopodobieństwo wyprowadzenia* l definiujemy jako

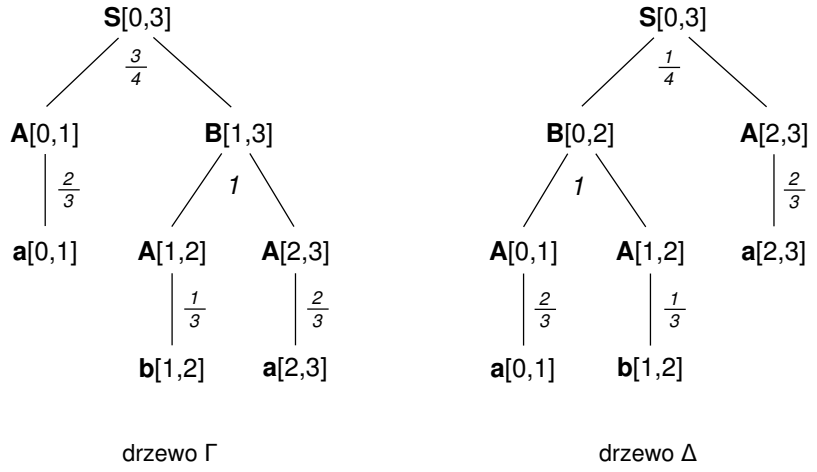
$$\mathbf{P}(l) := P(r_1) \cdot P(r_2) \cdots P(r_m) = \prod_{i=1}^{m} P(r_i).$$

Innymi słowy, prawdopodobieństwo wyprowadzenia to iloczyn prawdopodobieństw wszystkich produkcji użytych w wyprowadzeniu.

Definicja 4.3 (prawdopodobieństwo wyprowadzenia lewostronnego)**.** Ponieważ wyprowadzenie lewostronne jest wyprowadzeniem, więc *prawdopodobieństwo wyprowadzenia lewostronnego* definiujemy identycznie jak prawdopodobieństwo wyprowadzenia, tj. jako iloczyn prawdopodobieństw wszystkich produkcji użytych w tym wyprowadzeniu lewostronnym.

Definicja 4.4 (prawdopodobieństwo drzewa wyprowadzenia)**.** *Prawdopodobieństwo drzewa wyprowadzenia* definiujemy jako iloczyn prawdopodobieństw wszystkich produkcji użytych w tym drzewie wyprowadzenia.

Jeżeli istnieje wyprowadzenie lewostronne odpowiadające danemu drzewu wyprowadzenia, to prawdopodobieństwo tego drzewa i prawdopodobieństwo tego wyprowadzenia są sobie równe.



Rysunek 4.1. Przykładowe drzewa wyprowadzenia łańcucha *aba*. Zaznaczono prawdopodobieństwa produkcji.

W szczególności prawdopodobieństwo drzewa wyprowadzenia łańcucha symboli terminalnych jest równe prawdopodobieństwu wyprowadzenia lewostronnego tego łańcucha symboli terminalnych.

Jako iloczyny nieujemnych prawdopodobieństw produkcji, prawdopodobieństwa wyprowadzenia, wyprowadzenia lewostronnego i drzewa rozkładu są zawsze nieujemne.

Przykład 4.2 (prawdopodobieństwo drzewa wyprowadzenia)**.** Niech dana będzie probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$ z przykładu 4.1 oraz łańcuch *aba*. Na rysunku 4.1 przedstawiono dwa drzewa rozkładu tego łańcucha wraz z prawdopodobieństwami użytych w nich produkcji. Drzewu Γ odpowiada wyprowadzenie lewostronne

$$S \Rightarrow AB \Rightarrow aB \Rightarrow aAA \Rightarrow abA \Rightarrow aba,$$

zaś drzewu Δ — wyprowadzenie lewostronne

$$S \Rightarrow BA \Rightarrow AAA \Rightarrow aAA \Rightarrow abA \Rightarrow aba.$$

Prawdopodobieństwo drzewa Γ to

$$\mathbf{P}(\Gamma) = P(S \to AB)\, P(A \to a)\, P(B \to AA)\, P(A \to b)\, P(A \to a) = \\ = \tfrac{3}{4} \cdot \tfrac{2}{3} \cdot 1 \cdot \tfrac{1}{3} \cdot \tfrac{2}{3} = \tfrac{1}{9}.$$

Prawdopodobieństwo drzewa Δ to

$$\mathbf{P}(\Delta) = P(S \to BA)\, P(B \to AA)\, P(A \to a)\, P(A \to b)\, P(A \to a) = \\ = \tfrac{1}{4} \cdot 1 \cdot \tfrac{2}{3} \cdot \tfrac{1}{3} \cdot \tfrac{2}{3} = \tfrac{1}{27}.$$

Definicja 4.5 (prawdopodobieństwo wyprowadzalności łańcucha z symbolu)**.** Niech dane będą: probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$, łańcuch $\omega \in (V \cup T)^+$ oraz symbol nieterminalny $A \in V$. *Prawdopodobieństwo wyprowadzalności łańcucha ω z symbolu A* definiujemy jako sumę prawdopodobieństw wszystkich drzew wyprowadzenia o korzeniu A i plonie ω. Prawdopodobieństwo to oznaczamy przez $\mathbf{P}(\omega|A)$.

Prawdopodobieństwo wyprowadzalności łańcucha z symbolu jest zawsze nieujemne, ponieważ jest sumą nieujemnych prawdopodobieństw wyprowadzeń lewostronnych.

Definicja 4.6 (prawdopodobieństwo łańcucha)**.** Niech dane będą probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$ i łańcuch $\omega \in (V \cup T)^*$. *Prawdopodobieństwo łańcucha ω* definiujemy jako sumę prawdopodobieństw wszystkich drzew rozkładu łańcucha ω i oznaczamy przez $\mathbf{P}(\omega)$.

Prawdopodobieństwo każdego łańcucha jest nieujemne, ponieważ jest sumą nieujemnych prawdopodobieństw drzew rozkładu.

Prawdopodobieństwo łańcucha symboli terminalnych $w \in T^+$ jest sumą prawdopodobieństw wszystkich wyprowadzeń lewostronnych łańcucha w z symbolu początkowego S gramatyki G.

Prawdopodobieństwo łańcucha $\omega \in (V \cup T)^+$ jest równe prawdopodobieństwu wyprowadzalności łańcucha ω z symbolu początkowego S gramatyki $G = (V, T, R, S, P)$:

$$\mathbf{P}(\omega) = \mathbf{P}(\omega|S).$$

Przykład 4.3 (prawdopodobieństwo łańcucha)**.** Niech dana będzie probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$ z przykładu 4.1 oraz łańcuch *aba*. Przyjmijmy, że jedynymi drzewami wyprowadzenia tego łańcucha są Γ i Δ z przykładu 4.2. Wówczas

$$\mathbf{P}(aba) = \mathbf{P}(\Gamma) + \mathbf{P}(\Delta) = \tfrac{1}{9} + \tfrac{1}{27} = \tfrac{4}{27}.$$

Definicja 4.7. Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Wówczas przez $\hat{\mathcal{L}}(G)$ będziemy oznaczać zbiór wszystkich wyprowadzeń lewostronnych z symbolu początkowego S gramatyki G.

Definicja 4.8. Niech $G = (V, T, R, S)$ będzie gramatyką bezkontekstową. Wówczas przez $\mathcal{L}(G)$ będziemy oznaczać zbiór wszystkich lewostronnych wyprowadzeń łańcuchów złożonych z samych symboli końcowych gramatyki G.

Twierdzenie 4.2. *Zbiór $\hat{\mathcal{L}}(G)$ jest częściowo uporządkowany przez relację bycia podwyprowadzeniem.*

Dowód. Zwrotność, antysymetryczność i przechodniość relacji bycia podwyprowadzeniem wynika bezpośrednio z definicji 2.79. □

Twierdzenie 4.3. *Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową. Dla każdego skończonego maksymalnego antyłańcucha $K \subseteq \hat{\mathcal{L}}(G)$ zachodzi wówczas równość*

$$\sum_{l \in K} \mathbf{P}(l) \;=\; 1.$$

Dowód. Tezy twierdzenia dowiedziemy przez indukcję względem długości najdłuższego wyprowadzenia w antyłańcuchu. Jeżeli najdłuższe wyprowadzenie lewostonne w maksymalnym antyłańcuchu K ma długość 1, oznacza to, że antyłańcuch ten składa się ze wszystkich wyprowadzeń lewostronnych postaci

$$S \Rightarrow \omega,$$

gdzie $S \to \omega \in R$. Wówczas

$$\sum_{l \in K} \mathbf{P}(l) \;=\; \sum_{S \to \omega \in R} P(S \to \omega) \;=\; 1.$$

Przeprowadzimy teraz krok indukcyjny. Załóżmy, że dla każdego maksymalnego antyłańcucha K', którego najdłuższe wyprowadzenie ma długość m, zachodzi równość

$$\sum_{l \in K'} \mathbf{P}(l) \;=\; 1.$$

Niech K będzie maksymalnym antyłańcuchem, którego najdłuższe wyprowadzenie ma długość $m+1$. Niech to najdłuższe (bądź jedno z najdłuższych) wyprowadzenie lewostronne l będzie postaci

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow \zeta_m \Rightarrow \zeta_{m+1}.$$

Łańcuch ζ_m musi mieć postać

$$\zeta_m = uA\xi$$

dla pewnych $u \in T^*$, $A \in V$, $\xi \in (V \cup T)^*$. Łańcuch ζ_{m+1} musi zaś mieć postać

$$\zeta_{m+1} = u\omega\xi,$$

gdzie $\omega \in (V \cup T)^+$ jest takim łańcuchem, że istnieje produkcja $A \to \omega \in R$. Oznaczmy wyprowadzenie lewostronne

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow \zeta_m$$

przez l'.

Pokażemy teraz, że maksymalny antyłańcuch K musi zawierać wszystkie wyprowadzenia lewostronne l'' postaci

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow uA\xi \Rightarrow u\omega'\xi,$$

w której $A \to \omega' \in R$. Wiemy, że $l \in K$, zatem $l' \notin K$, ponieważ l' jest podwyprowadzeniem wyprowadzenia l. Gdyby istniało wyprowadzenie $l'' \notin K$ postaci

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow uA\xi \Rightarrow u\omega'\xi,$$

to zbiór $K \cup \{l''\}$ byłby maksymalnym antyłańcuchem, ponieważ z jednej strony wszystkie właściwe podwyprowadzenia wyprowadzenia l'' są jednocześnie podwyprowadzeniami wyprowadzenia l, a z drugiej — wyprowadzenie l'' nie jest podwyprowadzeniem żadnego innego wyprowadzenia lewostronnego z antyłańcucha K, bo ma długość równą długości najdłuższego wyprowadzenia antyłańcucha K. Wówczas jednak zbiór K nie mógłby być maksymalnym antyłańcuchem, co jest sprzeczne z naszym założeniem.

Zachodzi równość

$$\sum_{l'' \text{ jak wyżej}} \mathbf{P}(l'') \;=\; \sum_{A\to\omega' \in R} \mathbf{P}(l')\, P(A \to \omega') \;= \\ = \; \mathbf{P}(l') \underbrace{\sum_{A\to\omega' \in R} P(A \to \omega')}_{=1} \;=\; \mathbf{P}(l'). \quad (4.2)$$

Utwórzmy teraz zbiór K' przez usunięcie ze zbioru K wszytkich wyprowadzeń lewostronnych postaci

$$S = \zeta_0 \Rightarrow \zeta_1 \Rightarrow \ldots \Rightarrow uA\xi \Rightarrow u\omega'\xi,$$

i dołączenie zamiast nich wyprowadzenia l'. Zbiór K' jest oczywiście maksymalnym antyłańcuchem i

$$\sum_{l\in K'} \mathbf{P}(l) \;=\; \sum_{l\in K} \mathbf{P}(l)$$

z uwagi na zależność (4.2). Zauważmy też, że zbiór K' zawiera mniej wyprowadzeń o długości $m+1$ niż zbiór K.

Jeżeli długość najdłuższego wyprowadzenia w K' jest równa m, to wówczas równość

$$\sum_{l\in K} \mathbf{P}(l) \;=\; \sum_{l\in K'} \mathbf{P}(l) \;=\; 1$$

jest oczywista. W przeciwnym wypadku powtarzamy powyższą procedurę dopóty, dopóki nie otrzymamy maksymalnego antyłańcucha K'', w którym nie będzie już wyprowadzeń o długości $m+1$. Otrzymujemy wówczas

$$\sum_{l\in K} \mathbf{P}(l) \;=\; \sum_{l\in K''} \mathbf{P}(l) \;=\; 1,$$

co kończy dowód kroku indukcyjnego. □

Twierdzenie 4.4. *Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową. Dla każdego maksymalnego antyłańcucha $K \subseteq \hat{\mathcal{L}}(G)$ (nie musi być skończony) zachodzi wówczas nierówność*

$$\sum_{l \in K} \mathbf{P}(l) \leqslant 1.$$

Dowód. Przypuśćmy, że $K = \{l_1, l_2, \ldots\}$ jest przeliczalnym maksymalnym antyłańcuchem takim, że

$$\sum_{l \in K} \mathbf{P}(l) > 1$$

oraz

$$\mathbf{P}(l_1) \geqslant \mathbf{P}(l_2) \geqslant \ldots$$

Niech $K_n := \{l_1, l_2, \ldots, l_n\} \subset K$. Każdy ze zbiorów K_n jest skończonym antyłańcuchem (choć nie jest maksymalnym anyłańcuchem). Dla odpowiednio dużego n zachodziłoby wówczas

$$\sum_{l \in K_n} \mathbf{P}(l) > 1,$$

co jest sprzeczne z twierdzeniem 4.3, ponieważ każdy skończony antyłańcuch wyprowadzeń z $\hat{\mathcal{L}}(G)$ jest podzbiorem pewnego skończonego maksymalnego antyłańcucha. □

Twierdzenie 4.5. *Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową. Wówczas*

$$\sum_{l \in \mathcal{L}(G)} \mathbf{P}(l) \leqslant 1.$$

Dowód. Zbiór $\mathcal{L}(G)$ jest antyłańcuchem w $\hat{\mathcal{L}}(G)$. Istnieje zatem maksymalny antyłańcuch K taki, że $\mathcal{L}(G) \subseteq K$. Stąd

$$\sum_{l \in \mathcal{L}(G)} \mathbf{P}(l) \leqslant \sum_{l \in K} \mathbf{P}(l) \leqslant 1.$$

□

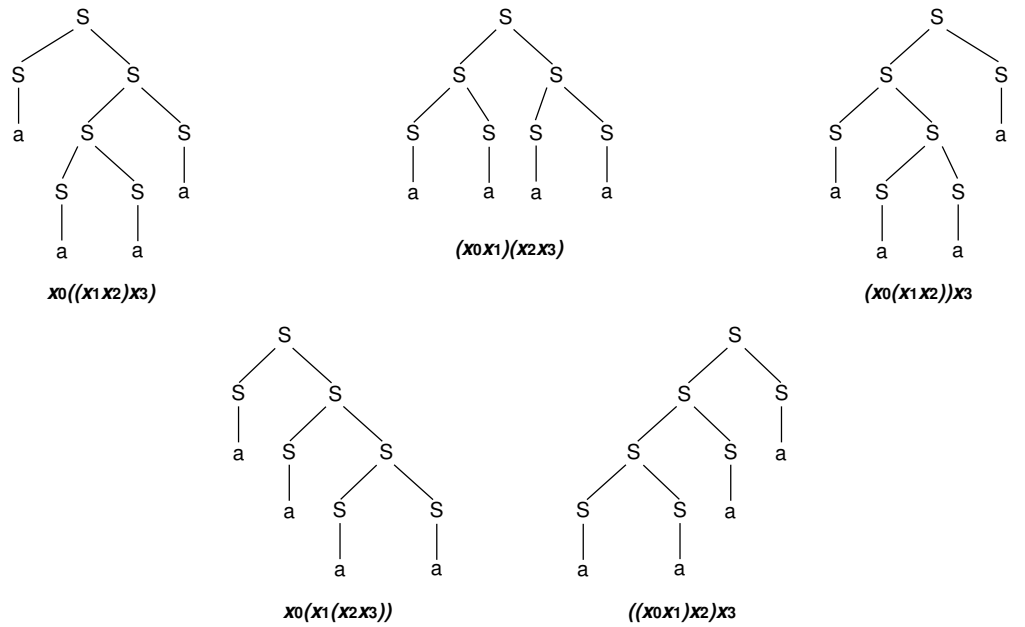
Twierdzenie 4.6. *Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową. Wówczas suma prawdopodobieństw wszystkich łańcuchów języka $L(G)$ jest nie większa od* 1. *Innymi słowy:*

$$\sum_{w \in L(G)} \mathbf{P}(w) \leqslant 1.$$

Dowód. Prawdopodobieństwo łańcucha to suma prawdopodobieństw wszystkich jego wyprowadzeń lewostronnych. Zbiór $\mathcal{L}(G)$ składa się ze wszystkich wyprowadzeń lewostronnych wszystkich łańcuchów języka $L(G)$, zatem

$$\sum_{w \in L(G)} \mathbf{P}(w) = \sum_{l \in \mathcal{L}(G)} \mathbf{P}(l) \leqslant 1.$$

□



Rysunek 4.2. Drzewa wyprowadzenia łańcucha *aaaa* i odpowiadające im rozmieszczenia nawiasów w iloczynie $x_0x_1x_2x_3$.

Twierdzenie 4.7. *Jeżeli $G = (V, T, R, S, P)$ jest probabilistyczną gramatyką bezkontekstową, to funkcja*

$$\mathbf{P}\colon L(G) \cup \{\infty\} \to \mathbb{R},$$
$$\mathbf{P}(w) \text{ jest prawdopodobieństwem łańcucha } w \text{ dla } w \in L(G),$$
$$\mathbf{P}(\infty) := 1 - \sum_{w \in L(G)} \mathbf{P}(w),$$

jest rozkładem prawdopodobieństwa na $L(G) \cup \{\infty\}$.

Dowód. Wiemy już, że $\mathbf{P}(w) \geqslant 0$ dla każdego $w \in L(G)$. Wiemy też, że

$$\sum_{w \in L(G)} \mathbf{P}(w) \;\leqslant\; 1.$$

Dlatego $\mathbf{P}(\infty) \geqslant 0$ oraz

$$\sum_{w \in L(G) \cup \{\infty\}} \mathbf{P}(w) \;=\; \sum_{w \in L(G)} \mathbf{P}(w) + \mathbf{P}(\infty) \;=\; 1.$$

□

Przykład 4.4. Rozpatrzmy następującą probabilistyczną gramatykę bezkontekstową $G = (V, T, R, S, P)$, która powstała z gramatyki bezkontekstowej z przykładu 2.7:

— $V = \{S\}$,
— $T = \{a\}$,

— $R = \{S \to SS,\, S \to a\}$,
— $S \in V$ jest symbolem początkowym.
— wartości prawdopodobieństw reguł:
$P(S \to SS) = \frac{2}{3}$,
$P(S \to a) = \frac{1}{3}$.

Gramatyka ta generuje język $L(G) = \{a, aa, aaa, \ldots\}$.

Rozpatrzmy iloczyn $x_0 x_1 \cdots x_n$, $n \geqslant 0$. Liczba różnych możliwych rozmieszczeń nawiasów w tym iloczynie tak, aby kolejność mnożeń była jednoznacznie wyznaczona, równa jest n-tej liczbie Catalana[1]:

$$C_n := \frac{1}{n+1}\binom{2n}{n}.$$

Istnieje bijekcja między rozmieszczeniami nawiasów w iloczynie $x_0 x_1 \cdots x_n$ a drzewami wyprowadzenia łańcucha $\underbrace{a \ldots a}_{n+1}$. Polega ona na stosowaniu produkcji $S \to SS$ do wyprowadzenia łańcucha $\underbrace{a \ldots a}_{n+1}$ zgodnie z kolejnością analogiczną do kolejności mnożeń wyznaczonej przez nawiasy w iloczynie $x_0 x_1 \cdots x_n$ (patrz rys. 4.2). Dlatego liczba różnych drzew wyprowadzenia łańcucha $\underbrace{a \ldots a}_{n+1}$ równa jest C_n (czyli liczba różnych drzew wyprowadzenia łańcucha $\underbrace{a \ldots a}_{n}$ równa jest C_{n-1}).

Każde z drzew rozkładu łańcucha $\underbrace{a \ldots a}_{n}$ korzysta n razy z produkcji $S \to a$ oraz $n-1$ razy z produkcji $S \to SS$.

Korzystając z tych faktów, możemy obliczyć $\sum_{w \in L(G)} \mathbf{P}(w)$:

$$\begin{aligned}
\sum_{w \in L(G)} \mathbf{P}(w) &= \sum_{n=1}^{\infty} \mathbf{P}(\underbrace{a \ldots a}_{n}) = \\
&= \sum_{n=1}^{\infty} C_{n-1} \cdot \mathbf{P}(\text{drzewo wyprowadzenia łańcucha } \underbrace{a \ldots a}_{n}) = \\
&= \sum_{n=1}^{\infty} C_{n-1} \cdot P(S \to a)^n \cdot P(S \to SS)^{n-1} = \\
&= \sum_{n=1}^{\infty} C_{n-1} \left(\tfrac{1}{3}\right)^n \left(\tfrac{2}{3}\right)^{n-1} = \tfrac{1}{3} \sum_{n=0}^{\infty} C_n \left(\tfrac{1}{3}\right)^n \left(\tfrac{2}{3}\right)^n = \\
&= \tfrac{1}{3} \sum_{n=0}^{\infty} C_n \left(\tfrac{2}{9}\right)^n = \tfrac{1}{3} \frac{1-\sqrt{1-4\cdot\frac{2}{9}}}{2\cdot\frac{2}{9}} = \tfrac{1}{2} < 1.
\end{aligned}$$

Jak widać, suma prawdopodobieństw wszystkich łańcuchów języka może być istotnie mniejsza od 1.

[1] Według [3].

4.3. Prawdopodobieństwo zewnętrzne i wewnętrzne

Definicja 4.9 (Prawdopodobieństwo zewnętrzne)**.** Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową, $w = w_1 \dots w_n \in L(G)$, $w_1, \dots, w_n \in T$. *Prawdopodobieństwo zewnętrzne* $\alpha(A, i, j)$ dla łańcucha w definiujemy jako

$$\alpha(A, i, j) = \alpha_w(A, i, j) := \mathbf{P}(w_1 \dots w_i A w_{j+1} \dots w_n).$$

Definicja 4.10 (Prawdopodobieństwo wewnętrzne)**.** Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową, $w = w_1 \dots w_n \in L(G)$, $w_1, \dots, w_n \in T$. *Prawdopodobieństwo wewnętrzne* $\beta(A, i, j)$ dla łańcucha w definiujemy jako

$$\beta(A, i, j) = \beta_w(A, i, j) := \mathbf{P}(w_{i+1} \dots w_j | A).$$

Prawdopodobieństwa wewnętrzne i zewnętrzne symboli definiuje się w celu ułatwienia obliczeń związanych z danym łańcuchem symboli terminalnych $w \in L(G)$. Dlatego w dalszych rozważaniach będzie na ogół jasne, o jakim łańcuchu mowa; będę wówczas pomijał oznaczenie łańcucha i pisał krótko $\alpha(A, i, j)$, $\beta(A, i, j)$ zamiast $\alpha_w(A, i, j)$, $\beta_w(A, i, j)$.

W dalszej części pracy będę również czasami zamiast symboli gramatyki (A) używał etykiet odpowiadających im wierzchołków drzewa wyprowadzenia ($A[i, j]$). Należy wówczas zapis $A[i, j]$ traktować jako symbol A, zaś parę liczb $[i, j]$ jedynie jako dodatkową informację, ułatwiającą interpretację wzoru. Na przykład zapisy

$$\beta(A, i, j) := \mathbf{P}(w_{i+1} \dots w_j | A[i, j])$$

i

$$\beta(A, i, j) := \mathbf{P}(w_{i+1} \dots w_j | A)$$

są równoważne i znaczą dokładnie to samo.

4.4. Algorytmy efektywnego obliczania prawdopodobieństwa łańcucha

4.4.1. Obliczanie prawdopodobieństwa łańcucha

Sumowanie prawdopodobieństw wszystkich możliwych drzew wyprowadzeń nie jest na ogół efektywną metodą obliczania prawdopodobieństwa danego łańcucha — możliwych drzew rozkładu, zwłaszcza dla długich łańcuchów, może być bardzo dużo. Istnieją wydajniejsze algorytmy obliczania

prawdopodobieństwa łańcucha, które wykorzystują prawdopodobieństwa zewnętrzne i wewnętrzne.

W dalszej części pracy będę przyjmował, że dana gramatyka jest w postaci normalnej Chomsky'ego. Nie jest to istotne ograniczenie, ponieważ dla każdej probabilistycznej gramatyki bezkontekstowej można znaleźć równoważną gramatykę w takiej postaci, jak wiemy z twierdzenia 2.18.

Ze względów technicznych będę dopuszczał stosowanie zapisu $P(r)$ dla dowolnych produkcji r, także takich, które nie występują w zbiorze reguł rozważanej gramatyki. Będziemy wówczas przyjmować $P(r) = 0$ dla dowolnej produkcji $r \notin R$.

4.4.2. Algorytm wewnętrzny

Algorytm wewnętrzny oblicza prawdopodobieństwo łańcucha korzystając z prawdopodobieństw wewnętrznych:

$$\mathbf{P}(w_1 \dots w_n) = \mathbf{P}(w_1 \dots w_n | S[0,n]) = \beta(S, 0, n).$$

Algorytm wykorzystuje indukcję względem drzewa wyprowadzenia łańcucha. Obliczenia są przeprowadzane wstępująco — najpierw obliczamy prawdopodobieństwa wewnętrzne reguł końcowych, potem reguł, z których zostały wyprowadzone, i tak dalej, aż do symbolu początkowego.

Ponieważ gramatyka jest w postaci normalnej Chomsky'ego, więc reguły końcowe muszą być produkcjami postaci $A \to w_k$. Dzięki temu łatwo obliczymy wartości prawdopodobieństwa wewnętrznego dla symboli preterminalnych (czyli poprzedzających symbole terminalne):

$$\beta(A, k-1, k) = \mathbf{P}(w_k | A[k-1,k]) = P(A \to w_k).$$

Aby obliczyć $\beta(A, i, j)$, postępujemy następująco. Ponieważ gramatyka jest w postaci normalnej Chomsky'ego, pierwsza produkcja musi być postaci $A \to BC$ dla pewnych zmiennych B, C. Wówczas zachodzą następujące równości:

$$\begin{aligned}
\beta(A,i,j) &= \mathbf{P}(w_{i+1} \dots w_j | A[i,j]) = \\
&= \sum_{B,C} \sum_{k=i+1}^{j-1} \mathbf{P}(B[i,k]C[k,j] | A[i,j]) \cdot \\
&\qquad \cdot \mathbf{P}(w_{i+1} \dots w_k | B[i,k]) \cdot \mathbf{P}(w_{k+1} \dots w_j | C[k,j]) = \\
&= \sum_{B,C} \sum_{k=i+1}^{j-1} P(A \to BC) \cdot \beta(B,i,k) \cdot \beta(C,k,j).
\end{aligned}$$

Przykład 4.5 (Algorytm wewnętrzny)**.** Niech będzie dana następująca probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$ w postaci normalnej Chomsky'ego:

— $V = \{S, A, B, C, D\}$,
— $T = \{a, b, c, d\}$,
— $R = \{S \to AB,\ S \to BA,\ B \to CD,\ A \to a,\ A \to b,\ C \to c,$
$D \to b,\ D \to d\}$,
— $S \in V$ jest symbolem początkowym,
— wartości prawdopodobieństw reguł:
$P(S \to AB) = 0.8$,
$P(S \to BA) = 0.2$,
$P(B \to CD) = 1$,
$P(A \to a) = 0.9$,
$P(A \to b) = 0.1$,
$P(C \to c) = 1$,
$P(D \to b) = 0.5$,
$P(D \to d) = 0.5$.

Niech dany będzie łańcuch *acb*, którego prawdopodobieństwo chcemy obliczyć. Na początku obliczamy prawdopodobieństwa wewnętrzne dla symboli preterminalnych:

$$\begin{aligned}
\beta(A, 0, 1) &= P(A \to a) = 0.9, \\
\beta(C, 1, 2) &= P(C \to c) = 1, \\
\beta(A, 2, 3) &= P(A \to b) = 0.1, \\
\beta(D, 2, 3) &= P(D \to b) = 0.5.
\end{aligned}$$

Następnie obliczamy prawdopodobieństwa wewnętrzne symboli z coraz szerszymi zakresami:

$$\begin{aligned}
\beta(B, 1, 3) &= P(B \to CD) \cdot \beta(C, 1, 2) \cdot \beta(D, 2, 3) = \\
&= 1 \cdot 1 \cdot 0.5 = 0.5, \\
\beta(S, 0, 3) &= P(S \to AB) \cdot \beta(A, 0, 1) \cdot \beta(B, 1, 3) = \\
&= 0.8 \cdot 0.9 \cdot 0.5 = 0.45.
\end{aligned}$$

Stąd ostatecznie otrzymujemy prawdopodobieństwo całego łańcucha:

$$\mathbf{P}(acb) = \beta(S, 0, 3) = 0.45.$$

Całą procedurę można w skrócie przedstawić w tabelce, której wiersze etykietowane są początkami zakresów symboli, a kolumny — końcami zakresów:

	1	2	3
0	$\beta(A,0,1)=0.9$		$\beta(S,0,3)=0.8\cdot 0.9\cdot 0.5=0.45$
1		$\beta(C,1,2)=1$	$\beta(B,1,3)=1\cdot 1\cdot 0.5=0.5$
2			$\beta(A,2,3)=0.1$ $\beta(D,2,3)=0.5$
	a	c	b

4.4.3. Algorytm zewnętrzny

Algorytm zewnętrzny jest procedurą zstępującą, wykorzystuje prawdopodobieństwa zewnętrzne i wewnętrzne oraz indukcję względem drzewa wyprowadzenia zdania. Wykorzystywana jest następująca równość:

$$\begin{aligned}\mathbf{P}(w_1\dots w_n) &= \\ &= \sum_A \mathbf{P}(w_1\dots w_{k-1}A[k-1,k]w_{k+1}\dots w_n)\cdot\mathbf{P}(w_k|A[k-1,k]) = \\ &= \sum_A \alpha(A,k-1,k)\cdot P(A\to w_k).\end{aligned}$$

Na początku rozważmy, jakie jest prawdopodobieństwo zewnętrzne dla symboli nieterminalnych, których zakres rozciąga się na cały analizowany łańcuch:

— dla symbolu początkowego S:

$$\alpha(S,0,n)=1,$$

— dla pozostałych zmiennych $A\in V\setminus\{S\}$:

$$\alpha(A,0,n)=0.$$

Krok indukcyjny: rozważany węzeł A może mieć brata po prawej lub po lewej stronie. Sumujemy obie możliwości:

$$\begin{aligned}\alpha(A,i,j) &= \sum_{B,C}\sum_{k=j+1}^{n}\mathbf{P}(w_1\dots w_iC[i,k]w_{k+1}\dots w_n)\cdot \\ &\quad\cdot\mathbf{P}(A[i,j]B[j,k]|C[i,k])\cdot\mathbf{P}(w_{j+1}\dots w_k|B[j,k])+ \\ &\quad+\sum_{B,C}\sum_{k=j+1}^{n}\mathbf{P}(w_1\dots w_kC[k,j]w_{j+1}\dots w_n)\cdot \\ &\quad\cdot\mathbf{P}(B[k,i]A[i,j]|C[k,j])\cdot\mathbf{P}(w_{k+1}\dots w_i|B[k,i]) = \\ &= \sum_{B,C}\sum_{k=j+1}^{n}\alpha(C,i,k)\cdot P(C\to AB)\cdot\beta(B,j,k)+ \\ &\quad+\sum_{B,C}\sum_{k=j+1}^{n}\alpha(C,k,j)\cdot P(C\to BA)\cdot\beta(B,k,i).\end{aligned}$$

4.5. Drzewo Viterbiego

Definicja 4.11 (drzewo Viterbiego)**.** Niech $G = (V, T, R, S, P)$ będzie probabilistyczną gramatyką bezkontekstową, niech łańcuch $w \in L(G)$. *Drzewem Viterbiego* łańcucha w nazywamy to drzewo wyprowadzenia łańcucha w, które ma największe prawdopodobieństwo spośród wszystkich drzew rozkładu tego łańcucha.

Do znajdowania drzewa Viterbiego można użyć zmodyfikowanych algorytmów obliczania prawdopodobieństwa łańcucha, w szczególności tych wykorzystujących prawdopodobieństwa zewnętrzne i wewnętrzne.

Algorytm znajdowania drzewa Viterbiego przebiega następująco. Niech dany będzie łańcuch $w = w_1 \dots w_n \in L(G)$. Drzewo wyprowadzenia łańcucha $w_{a+1} \dots w_b$ z symbolu $A[a, b]$ o największym prawdopodobieństwie będziemy oznaczać przez $\Psi(A[a, b])$, zaś prawdopodobieństwo tego drzewa przez $\delta(A[a, b])$.

Na początku symbolom preterminalnym przypisujemy prawdopodobieństwa ich unarnych produkcji:

$$\delta(A[k, k]) = P(A \to w_k),$$

oraz związane z tymi produkcjami drzewa rozkładu:

$$\Psi(A[k, k]) = (A,\ A \to w_k\,[k, k],\ w_k).$$

Dalej postępujemy podobnie jak w przypadku algorytmu wewnętrznego, ale zamiast sumować prawdopodobieństwa wewnętrzne, znajdujemy maksimum prawdopodobieństw:

$$\delta(A[a, b]) = \max_{B,C;\, a<c<b} P(A \to BC) \cdot \delta(B[a, c]) \cdot \delta(C[c, b]).$$

Zapisujemy też, które drzewo wyprowadzenia dało to największe prawdopodobieństwo:

$$\Psi(A[a, b]) = \operatorname*{arg\,max}_{l=(A,A\to BC\dots)} P(A \to BC) \cdot \delta(B[a, c]) \cdot \delta(C[c, b]).$$

Ostatecznie drzewem Viterbiego łańcucha w jest $\Psi(S[0, n])$, a jego prawdopodobieństwo jest równe $\delta(S[0, n])$.

4.6. Uczenie gramatyki

W zastosowaniach praktycznych probabilistycznych gramatyk bezkontekstowych ważne jest, żeby model matematyczny dobrze oddawał rzeczywiste

własności języka. Dlatego istotne było opracowanie metod, które pozwalają dobrać prawdopodobieństwa reguł tak, aby obliczone na ich podstawie prawdopodobieństwa poszczególnych łańcuchów (zdań) były jak najbliższe rzeczywistym częstościom występowania zdań w danym korpusie.

Przyjmijmy, że mamy daną gramatykę bezkontekstową $G = (V, T, R, S)$, której regułom chcielibyśmy przypisać prawdopodobieństwa. Dysponujemy również korpusem złożonym z pewnej (najlepiej dużej) liczby zdań.

Procedura przypisywania gramatyce prawdopodobieństw reguł na podstawie danych z korpusu to właśnie *uczenie gramatyki*. W celu uczenia probabilistycznej gramatyki bezkontekstowej stosuje się *algorytm wewnętrzno-zewnętrzny*, wykorzystujący prawdopodobieństwa wewnętrzne i zewnętrzne.

Oznaczmy przez $C(r)$ liczbę, ile razy reguła r została użyta w korpusie. To, co potrzebujemy wyznaczyć, to wartość

$$\hat{P}(A \to w) := \frac{C(A \to w)}{\sum_{u \in (V \cup T)^*} C(A \to u)}.$$

Wartość tę możemy obliczyć bezpośrednio z powyższego wzoru, jeżeli zdania z korpusu zostały *sparsowane*, czyli poddane analizie składniowej — znamy ich wyprowadzenia. W przeciwnym wypadku — jeżeli mamy do dyspozycji same zdania — posługujemy się następującą metodą.

Na początek przypisujemy regułom pewne arbitralnie dobrane (mogą być losowe) prawdopodobieństwa. Następnie obliczamy prawdopodobieństwa zdań na podstawie tych prawdopodobieństw reguł i porównujemy z rzeczywistymi częstościami wystąpienia tych zdań. W zależności od wyniku porównania, wartości prawdopodobieństw reguł aktualizujemy tak, aby zmaksymalizować prawdopodobieństwo korpusu.

4.7. Algorytm Cocke'a-Youngera-Kasamiego jako przykład algorytmu parsowania probabilistycznych gramatyk bezkontekstowych[2]

Algorytm Cocke'a-Youngera-Kasamiego jest algorytmem parsowania gramatyk bezkontekstowych — służy do znajdywania drzewa wyprowadzenia danego łańcucha symboli terminalnych. Do znalezienia drzewa Viterbiego łańcucha dla danej probabilistycznej gramatyki bezkontekstowej używa się zmodyfikowanego algorytmu CYK.

Algorytm CYK korzysta z faktu, że aby znaleźć drzewo rozkładu łańcucha $w = uv$ można najpierw znaleźć drzewa rozkładu dla łańcuchów u i v (dla

[2]Na podstawie [4].

wszystkich możliwych takich rozkładów $w = uv$), a następnie odpowiednio połączyć znalezione drzewa.

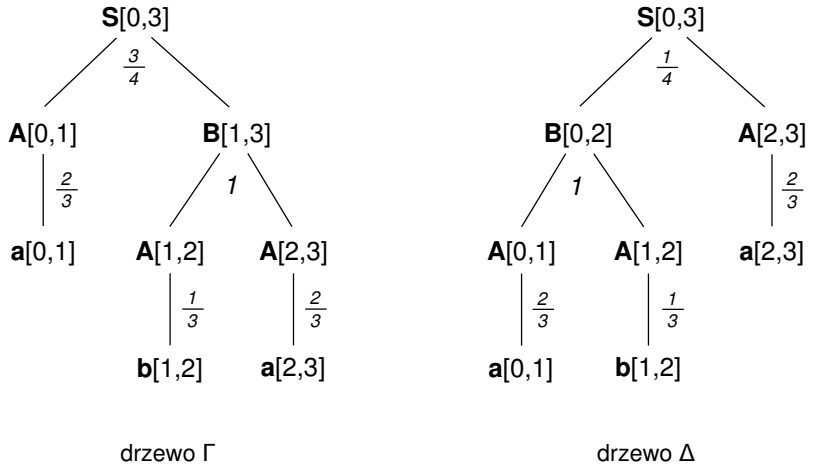
Algorytm ten przypomina algorytm wewnętrzny opisany w sekcji 4.4 służący do obliczania prawdopodobieństwa łańcucha. Tym razem jednak zamiast w każdym kroku sumować prawdopodobieństwa, wybieramy największe spośród nich.

Dla probabilistycznej gramatyki bezkontekstowej $G = (V, T, R, S)$ i łańcucha $w = w_1 \dots w_n \in T^+$ algorytm wykorzystuje tablicę t o wymiarach $|w| \times |w| \times |V|$, czyli $n \times n \times |V|$ W komórce $t_{i,j,A}$ przechowywana jest wartość największego prawdopodobieństwa drzewa wyprowadzenia o korzeniu A i plonie $w_i \dots w_j$.

Dodatkowa tablica t' o tych samych wymiarach przechowuje wskaźniki, za pomocą których można odtworzyć drzewo, którego prawdopodobieństwo zostało obliczone jako prawdopodobieństwo Viterbiego.

Algorytm 4.1 (algorytm CYK znajdowania drzewa Viterbiego dla probabilistycznych gramatyk bezkontekstowych)**.** Niech dana będzie probabilistyczna gramatyka bezkontekstowa $G = (V, T, R, S, P)$ oraz łańcuch $w = w_1 \dots w_n$.

1. Utwórz pustą tablicę t o wymiarach $n \times n \times |V|$.
2. Dla i od 1 do n wykonaj:
   — dla wszystkich symboli pomocniczych $A \in V$ wykonaj:
      — jeżeli istnieje reguła $A \to w_i \in R$, to:
         a) $t_{i,i,A} := P(A \to w_i)$,
         b) w komórce $t'_{i,i,A}$ tablicy wskaźników zaznacz, że jest to liść drzewa wyprowadzenia.
3. Dla i od 2 do n wykonaj:
   — dla j od 1 do $n - i + 1$ wykonaj:
      — dla k od 1 do $i - 1$ wykonaj:
         — dla wszystkich trójek symboli pomocniczych (A, B, C) wykonaj:
            a) $p := t_{j,j+k-1,B} \cdot t_{j+k,j+i-1,C} \cdot P(A \to BC)$;
            b) jeśli $p > t_{j,j+i-1,A}$, to:
               i. $t_{j,j+i-1,A} := p$,
               ii. w komórce $t'_{j,j+i-1,A}$ tablicy wskaźników zapisz wskaźniki do komórek $t'_{j,j+k-1,B}$ i $t'_{j+k,j+i-1,C}$.
4. Po zakończeniu działania algorytmu prawdopodobieństwo Viterbiego łańcucha w znajduje się w komórce $t_{1,n,S}$. Wyprowadzenie to można odczytać, podążając za wskaźnikami znajdującymi się w komórce $t'_{1,n,S}$ tablicy wskaźników.



Rysunek 4.3. Dwa różne drzewa wyprowadzenia łańcucha *aba*. Prawdopodobieństwo drzewa Γ wynosi $\frac{1}{9}$. Prawdopodobieństwo drzewa Δ wynosi $\frac{1}{27}$. Drzewem Viterbiego łańcucha *aba* jest zatem drzewo Γ.

Przykład 4.6 (algorytm CYK dla probabilistycznych gramatyk bezkontekstowych)**.** Niech dana będzie probabilistyczna gramatyka bezkontekstowa w postaci normalnej Chomsky'ego z przykładu 4.1 na stronie 41.

Niech dany będzie łańcuch wejściowy *aba*, który posiada dwa różne drzewa wyprowadzenia (rys. 4.3).

Na początku algorytmu tworzona jest pusta trójwymiarowa tablica t o wymiarach $3 \times 3 \times 3$:

	1 (S,A,B)	2 (S,A,B)	3 (S,A,B)
1	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,—,—)
2	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,—,—)
3	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,—,—)
	a	b	a

Znajdujemy prawdopodobieństwa produkcji, po prawej stronie których występują symbole końcowe łańcucha wejściowego:

$$P(A \to a) = \frac{2}{3},$$
$$P(A \to b) = \frac{1}{3}.$$

Prawdopodobieństwa te zapisujemy w odpowiednich miejscach tablicy:

	1	2	3
	(S,**A**,B)	(S,**A**,B)	(S,**A**,B)
1	$(0, \mathbf{\frac{2}{3}}, 0)$	$(0,0,0)$	$(0,0,0)$
wskaźniki	(—,/,—)	(—,—,—)	(—,—,—)
2	$(0,0,0)$	$(0, \mathbf{\frac{1}{3}}, 0)$	$(0,0,0)$
wskaźniki	(—,—,—)	(—,/,—)	(—,—,—)
3	$(0,0,0)$	$(0,0,0)$	$(0, \mathbf{\frac{2}{3}}, 0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,/,—)
	a	b	a

W następnej kolejności szukamy symboli, z których można wyprowadzić podłańcuchy długości 2:

$$
\begin{aligned}
&B \to AA, \\
&\qquad p := \tfrac{2}{3} \cdot \tfrac{1}{3} \cdot P(B \to AA) = \tfrac{2}{3} \cdot \tfrac{1}{3} \cdot 1 = \tfrac{2}{9}; \\
&B \to AA, \\
&\qquad p := \tfrac{1}{3} \cdot \tfrac{2}{3} \cdot P(B \to AA) = \tfrac{1}{3} \cdot \tfrac{2}{3} \cdot 1 = \tfrac{2}{9}.
\end{aligned}
$$

Aktualizujemy zawartość tablicy, wprowadzając do niej znalezione symbole i obliczone wartości prawdopodobieństw, a także odpowiednie wskaźniki:

	1	2	3
	(S,*A*,B)	(S,*A*,**B**)	(S,*A*,**B**)
1	$(0, \mathbf{\frac{2}{3}}, 0)$	$(0,0, \mathbf{\frac{2}{9}})$	$(0,0,0)$
wskaźniki	(—,/,—)	(—,—,[**1,1,A;2,2,A**])	(—,—,—)
2	$(0,0,0)$	$(0, \mathbf{\frac{1}{3}}, 0)$	$(0,0, \mathbf{\frac{2}{9}})$
wskaźniki	(—,—,—)	(—,/,—)	(—,—,[**2,2,A;3,3,A**])
3	$(0,0,0)$	$(0,0,0)$	$(0, \mathbf{\frac{2}{3}}, 0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,/,—)
	a	b	a

Na koniec szukamy symboli, z których można wyprowadzić podłańcuchy długości 3 — w rozważanym przykładzie jest to długość całego łańcucha. Odnajdujemy dwie produkcje, po lewej stronie których występuje symbol początkowy S gramatyki:

$$
\begin{aligned}
&S \to AB, \\
&\qquad p_1 := \tfrac{2}{3} \cdot \tfrac{2}{9} \cdot P(S \to AB) = \tfrac{2}{3} \cdot \tfrac{2}{9} \cdot \tfrac{3}{4} = \tfrac{1}{9}; \\
&S \to BA, \\
&\qquad p_2 := \tfrac{2}{3} \cdot \tfrac{2}{9} \cdot P(S \to BA) = \tfrac{2}{3} \cdot \tfrac{2}{9} \cdot \tfrac{1}{4} = \tfrac{1}{27}.
\end{aligned}
$$

Do komórki $t_{1,3,S}$ wpisujemy wartość p_1, ponieważ $p_1 > p_2$. Wpisujemy też wskaźnik na poddrzewa o korzeniach w A i B:

	1	2	3
	(S,*A*,B)	(S,A,*B*)	(**S**,A,*B*)
1	$(0, \frac{2}{3}, 0)$	$(0, 0, \frac{2}{9})$	$(\mathbf{\frac{1}{9}}, 0, 0)$
wskaźniki	(—,/,—)	(—,—,[1,1,A;2,2,A])	(**[1,1,A;2,3,B]**,—,—)
2	$(0, 0, 0)$	$(0, \frac{1}{3}, 0)$	$(0, 0, \frac{2}{9})$
wskaźniki	(—,—,—)	(—,/,—)	(—,—,[2,2,A;3,3,A])
3	$(0, 0, 0)$	$(0, 0, 0)$	$(0, \frac{2}{3}, 0)$
wskaźniki	(—,—,—)	(—,—,—)	(—,/,—)
	a	*b*	*a*

Drzewo Viterbiego odczytujemy podążając za wskaźnikami z komórki $t_{1,3,S}$.

Lemat 4.8. *W algorytmie CYK dla probabilistycznych gramatyk bezkontekstowych w każdej komórce $t_{i,i+m-1,A}$ ($m \geqslant 1$) tablicy t znajduje się największe spośród prawdopodobieństw drzew rozkładu o korzeniu A i plonie $w_i \dots w_{i+m-1}$ (pod warunkiem, że takie drzewo istnieje). To drzewo o największym prawdopodobieństwie można odczytać podążając za wskaźnikami znajdującymi się w komórce $t'_{i,i+m-1,A}$ tablicy wskaźników t'.*

Dowód. Dowód przeprowadzimy metodą indukcji matematycznej ze względu na m.

Niech $m = 1$. Komórki $t_{i,i,A}$, $t'_{i,i,A}$ są wypełniane w początkowych krokach algorytmu CYK. Dla danego symbolu końcowego w_i i danego symbolu pomocniczego istnieje co najwyżej jedno drzewo wyprowadzenia o korzeniu A i plonie w_i. Jest to drzewo odpowiadające wyprowadzeniu (lewostronnemu) $A \Rightarrow w_i$, pod warunkiem, że istnieje produkcja $A \to w_i$. Prawdopodobieństwo tego drzewa równe jest prawdopodobieństwu reguły $P(A \to w_i)$. Algorytm CYK wpisuje w komórce $t_{i,i,A}$ wartość równą temu prawdopodobieństwu reguły i pozostaje ona niezmieniona aż do końca działania algorytmu. W komórce $t'_{i,i,A}$ algorytm wpisuje informację, że symbol w_i jest liściem drzewa wyprowadzenia. Szukane drzewo wyprowadzenia odczytujemy jako zawierające pojedynczą produkcję $A \to w_i$. Teza lematu jest zatem spełniona dla $m = 1$.

Wykonajmy teraz krok indukcyjny. Załóżmy, że teza lematu jest spełniona dla wszystkich $i \leqslant m$. Rozważmy komórki postaci $t_{j,j+m,A}$.

Niech Γ będzie drzewem rozkładu o korzeniu A i plonie $w_j \dots w_{j+m}$. Ponieważ gramatyka, z której korzystamy, jest w postacji normalnej Chomsky'ego, zatem pierwsza produkcja użyta w drzewie Γ jest postaci $A \to B_0 C_0$ dla pewnych symboli pomocniczych B_0 i C_0. Niech Δ_0 oznacza poddrzewo drzewa Γ zaczepione w B_0, zaś E_0 — poddrzewo drzewa Γ zaczepione w C_0. Spośród wszystkich par drzew Δ i E takich, że Δ ma korzeń B i plon $w_j \dots w_{k-1}$, zaś E

ma korzeń C i plon $w_k \dots w_{j+m}$, dla pewnych $1 \leqslant k \leqslant i-1$, $B, C \in V$ takich, że istnieje produkcja $A \to BC$, para drzew $\Delta = \Delta_0$ i $\mathrm{E} = \mathrm{E}_0$ maksymalizuje iloczyn

$$\mathbf{P}(\Delta) \cdot \mathbf{P}(\mathrm{E}) \cdot P(A \to BC).$$

W komórce $t_{j,j+m,A}$ obliczana jest wartość

$$\max_{1 \leqslant k \leqslant i-1,\ A \to BC} t_{j,j+k-1,B} \cdot t_{j+k,j_i-1,C} \cdot P(A \to BC),$$

a zatem, na mocy założenia indukcyjnego, wartość największego spośród prawdopodobieństw drzew o korzeniu A i plonie $w_j \dots w_{j+m}$. W komórce $t'_{j,j+m,A}$ umieszczany jest wskaźnik do komórek $t'_{j,j+k-1,B_0}$ i $t'_{j+k,j+i-1,C_0}$, co pozwala na odtworzenie drzewa, którego prawdopodobieństwo równe jest $t_{j,j+m,A}$, w następujący sposób. Budujemy drzewo o korzeniu A, którego synami są B_0 i C_0. Następnie w wierzchołku B_0 zaczepiamy drzewo uzyskane według wskazań z $t'_{j,j+k-1,B_0}$ (rekurencyjnie), a w wierzchołku C_0 zaczepiamy drzewo uzyskane według wskazań z $t'_{j+k,j+i-1,C_0}$ (również rekurencyjnie). □

Twierdzenie 4.9. *Algorytm CYK dla probabilistycznych gramatyk bezkontekstowych zawsze znajduje drzewo Viterbiego wejściowego łańcucha.*

Dowód. Na mocy lematu 4.8 w komórce $t_{1,n,S}$ znajduje się wartość największego spośród prawdopodobieństw drzew o korzeniu S i plonie $w_1 \dots w_n$, które to drzewo można odtworzyć na podstawie wskaźników z komórki $t'_{1,n,S}$. Drzewo to jest szukanym drzewem Viterbiego łańcucha $w_1 \dots w_n$. □

Bibliografia

[1] Jay Earley, "An efficient context-free parsing algorithm", [w:] *Communications of the Association for Computing Machinery* 13 (2), s. 94-102, 1970.

[2] John E. Hopcroft, Jeffrey D. Ullman, *Wprowadzenie do teorii automatów, języków i obliczeń*, Wydawnictwo Naukowe PWN, Warszawa 2003.

[3] Jerzy Jaworski, Zbigniew Palka, Jerzy Szymański, *Matematyka dyskretna dla informatyków. Część I: Elementy kombinatoryki*, Wydawnictwo Naukowe UAM, Poznań 2008.

[4] Daniel Jurafsky, James H. Martin, *Speech and Language Processing*, Prentice Hall, New Jersey 2000.

[5] Joanna Jędrzejowicz, Andrzej Szepietowski, *Języki, automaty, złożoność obliczeniowa*, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2008.

[6] John G. Kemeny, J. Laurie Snell, *Finite Markov Chains*, Van Nostrand, Princeton 1960.

[7] Christopher D. Manning, Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, London 1999.

[8] Azaria Paz, *Introduction to Probabilistic Automata*, Academic Press, New York & London 1971.

[9] Jay M. Ponte, W. Bruce Croft, "A language modeling approach to information retrieval", [w:] *Research and Development in Information Retrieval*, s. 275-281, 1998.

[10] Kenneth A. Ross, Charles R. B. Wright, *Matematyka dyskretna*, Wydawnictwo Naukowe PWN, Warszawa 2003.

[11] Fei Song, W. Bruce Croft, "A general language model for information retrieval", [w:] *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, s. 279-280, 1999.

[12] Andreas Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities", [w:] *Computational Linguistics* 21 (2), s. 165-202, 1995.

[13] Robin J. Wilson, *Wprowadzenie do teorii grafów*, Wydawnictwo Naukowe PWN, Warszawa 1998.

[14] Stefan Zubrzycki, *Wykłady z rachunku prawdopodobieństwa i statystyki matematycznej*, PWN, Warszawa 1966.

Spis oznaczeń

A, B, C, D	symbole pomocnicze gramatyki
$A_1, A_2, \ldots$	zdarzenia
E	zbiór krawędzi grafu
F	zbiór stanów końcowych automatu
$\mathcal{F}$	ciało przeliczalnie addytywne; zbiór zdarzeń
G	gramatyka
L	język
M	automat
O	notacja asymptotyczna
P	prawdopodobieństwo produkcji
$\mathbf{P}$	prawdopodobieństwo
Q	zbiór stanów automatu
R	zbiór produkcji gramatyki
$\mathbb{R}$	zbiór liczb rzeczywistych
S	symbol początkowy gramatyki
T	zbiór symboli końcowych gramatyki
U	zbiór wierzchołków grafu
V	alfabet; zbiór symboli pomocniczych gramatyki
W	waga krawędzi grafu
X, Y, Z	symbole końcowe bądź pomocnicze gramatyki
$X_1, X_2, \ldots$	zmienne losowe
a, b, c	symbole wejściowe automatu; symbole końcowe gramatyki
d	odległość w grafie
e	zdarzenia elementarne; krawędzie grafu
i, j, k, l, m, n	liczby naturalne
p	rozkład prawdopodobieństwa
q_0	stan początkowy automatu
$q_0, q_1, q_2, \ldots$	stany automatu
r	produkcje gramatyki
s, t	stany łańcucha Markowa

u, v, w	łańcuchy symboli końcowych gramatyki
u, v, x, y	wierzchołki grafu
w, x, y, z	łańcuchy symboli wejściowych automatu
$w_1, w_2, \ldots$	symbole końcowe gramatyki
Γ, Δ	grafy, drzewa
Σ	alfabet wejściowy automatu
Ω	przestrzeń zdarzeń elementarnych
α	prawdopodobieństwo zewnętrzne
β	prawdopodobieństwo wewnętrzne
δ	funkcja przejścia automatu
ϵ	łańcuch pusty
ζ, ξ, ω	łańcuchy symboli pomocniczych i końcowych gramatyki
ν	prawdopodobieństwo Viterbiego
$\emptyset$	zbiór pusty