6 Ćwiczenia numer 6 - CYK

Algorytm ten służy do sprawdzania czy dane słowo należy do języka opisywanego przez gramatykę, która jest bezkontekstowa (typu 2) oraz jest zapisana w postaci normalnej Chomsky'ego (Chomsky Normal Form CNF).

Pamiętajmy, że algorytm ten działa tylko dla gramatyk bezkontekstowych zapisanych w postaci normalnej Chomsky'ego, a więc musimy się na wstępie upewnić, że z taką mamy do czynienia.

6.1. Postać normalna Chomsky'ego

Przypomnijmy, że gramatyka bezkontekstowa to taka, która ma po lewej stronie dokładnie jeden terminal, a po prawej może mieć cokolwiek, natomiast postać normalna Chomsky'ego może zawiera tylko dwa rodzaje produkcji:

- $A \to BC$
- $A \to a$

A zatem w postaci takiej:

- Po lewej stronie produkcji zawsze występuje dokładnie jeden symbol nieterminalny
- Po prawej stronie produkcji mogą znajdować się albo dwa symbole nieterminalne albo tylko jeden symbol terminalny

Istnieje twierdzenie które mówi, że każdą gramatykę bezkontekstową można przekształcić do postaci normalnej Chomsky'ego.

6.2. Konwersja gramatyki do CNF

Konwersja składa się z dwóch głównych etapów, etap pierwszy to:

- eliminacja ε produkcji, produkcji jednostkowych oraz bezużytecznych produkcji

Po tym etapie gramatyka ma być w postaci w której:

- Produkcja jest postaci $A \to a$ lub
- Produkcja jest postaci gdzie z lewej strony jest jeden nieterminal, a po prawej stronie jest 2 lub więcej symboli (terminalnych lub nieterminalnych)

Drugi etap to:

- zamiana produkcji tak aby każda prawa strona, która ma dwa lub więcej symboli, zawierała tylko symbole nieterminalne
- podział produkcji, które mają długość 3 lub więcej w produkcje kaskadowe, każda prawą stroną składającą się tylko z dwóch symboli

6.2.1. Etap pierwszy

Etap pierwszy składa się z:

1. Eliminacja ε produkcji
2. Eliminacja produkcji jednostkowych
3. Eliminacja bezużytecznych symboli

UWAGA: Etapy te muszą być stosowane w kolejności takiej jak wypisana, inaczej może się okazać, że coś co eliminowaliśmy wcześniej jest obecne po zastosowaniu kolejnej eliminacji.

- Eliminacja ε produkcji

Tutaj po prostu eliminujemy z naszej gramatyki produkcje, w wyniku których dostajemy słowo puste. Jeśli jakaś produkcja jest postaci $A \to \varepsilon$, to eliminujemy ją z gramatyki. W przypadku gdy jest to jedyna produkcja dla zmiennej A sytuacja jest dość prosta, bo z każdej innej produkcji która zawiera A również usuwamy zmienną A.

Zastanówmy się jednak co w przypadku, gdy z tej zmiennej A możemy także uzyskać coś oprócz ε ? Przykładowo mamy dodatkowo produkcję $A \to BD|\varepsilon$

Wtedy musimy rozpatrzeć dwa przypadki, gdy zmienna A jest słowem pustym i gdy nim nie jest. W związku z tym przekształcamy wszystkie produkcje zawierające A na takie dwa przypadki, przykładowo produkcje $B \to DAC$, przekształcimy na dwie:

$$B \to DC$$
$$B \to DAC$$

Oczywiście produkcje $A \to \varepsilon$ usuwamy z produkcji tak aby A nie mogło być już puste.

Przykład z Hopcrofta:

$$S \to AB$$
$$A \to aAA \mid \epsilon$$
$$B \to bBB \mid \epsilon$$

Zamieniamy na:

$$S \to AB \mid A \mid B$$

$$A \to aAA \mid aA \mid aA \mid a$$

$$B \to bBB \mid bB \mid b$$

- Eliminacja produkcji jednostkowych

Produkcja jednostkowa to produkcja postaci $A \to B$, gdzie A oraz B są zmiennymi. Z takimi produkcjami radzimy sobie po prostu dodając alternatywy do odpowiedniej produkcji aż do czasu zniknięcie produkcji jednostkowych.

Przykład z Hopcrofta:

$$\begin{array}{lcl} I & \to & a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ F & \to & I \mid (E) \\ T & \to & F \mid T * F \\ E & \to & T \mid E + T \end{array}$$

However, unit productions can complicate certain proofs, and they also introduce extra steps into derivations that technically need not be there. For instance, we could expand the T in production $E \to T$ in both possible ways, replacing it by the two productions $E \to F \mid T * F$. That change still doesn't eliminate unit productions, because we have introduced unit production $E \to F$ that was not previously part of the grammar. Further expanding $E \to F$ by the two productions for F gives us $E \to I \mid (E) \mid T * F$. We still have a unit production; it is $E \to I$. But if we further expand this I in all six possible ways, we get:

$$E \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \mid (E) \mid T * F \mid E + T$$

UWAGA: Po zamianie nie usuwamy z gramatyki zmiennych, a jedynie produkcje jednostkowe więc ostatecznie po tym etapie mamy:

$$\begin{array}{l} E \to E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ T \to T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

- Eliminacja bezużytecznych symboli

Symbol bezużyteczny to taki z którego nie można wyprowadzić słowa końcowego lub taki który nie jest osiągalny. Taki symbol eliminujemy z produkcji, jeśli eliminujemy do z produkcji, to usuwamy całe wyrażenie, które zawierało ten symbol, przykład z Hopcrofta:

$$\begin{array}{l} S \to AB \mid a \\ A \to b \end{array}$$

Z symbolu B nie możemy nic już wygenerować więc usuwamy go z produkcji:

$$\begin{array}{l} S \to a \\ A \to b \end{array}$$

Teraz symbol A nie jest osiągalny więc usuwamy produkcje i pozostaje:

$$S \to a.$$

6.2.2. Drugi etap

- zamiana produkcji tak aby każda prawa strona, która ma dwa lub więcej symboli, zawierała tylko symbole nieterminalne

Dla każdego terminala, który znajduje się po prawej stronie w ciągu o długości dwa lub dłuższym tworzymy osobą zmienną, zmienna taka będzie miała tylko jedną produkcję $Z \to t$.

Przykład z Hopcrofta, dla gramatyki po pierwszym etapie:

$$\begin{array}{l} E \to E+T \mid T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ T \to T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\ I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{array}$$

Tworzymy nowe zmienne:

$$\begin{array}{lcl} E & \to & EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\ T & \to & TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\ F & \to & LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\ I & \to & a \mid b \mid IA \mid IB \mid IZ \mid IO \\ A & \to & a \\ B & \to & b \\ Z & \to & 0 \\ O & \to & 1 \\ P & \to & + \\ M & \to & * \\ L & \to & (\\ R & \to &) \end{array}$$

- podział produkcji, które mają długość 3 lub więcej w produkcje kaskadowe, każda prawą stroną składającą się tylko z dwóch symboli

Dla każdego ciągu zostawiamy zmienną najbardziej po lewej stronie, a pozostałe zastępujemy zmienną itd. aż do otrzymania par.

> Now, all productions are in Chomsky Normal Form except for those with the bodies of length 3: EPT, TMF, and LER. Some of these bodies appear in more than one production, but we can deal with each body once, introducing one extra variable for each. For EPT, we introduce new variable C_1, and replace the one production, $E \to EPT$, where it appears, by $E \to EC_1$ and $C_1 \to PT$.

$$\begin{array}{lll}
E & \rightarrow & EC_1 \mid TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
T & \rightarrow & TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
F & \rightarrow & LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO \\
I & \rightarrow & a \mid b \mid IA \mid IB \mid IZ \mid IO \\
A & \rightarrow & a \\
B & \rightarrow & b \\
Z & \rightarrow & 0 \\
O & \rightarrow & 1 \\
P & \rightarrow & + \\
M & \rightarrow & * \\
L & \rightarrow & (\\
R & \rightarrow &) \\
C_1 & \rightarrow & PT \\
C_2 & \rightarrow & MF \\
C_3 & \rightarrow & ER
\end{array}$$

6.3. Stosowanie algorytmu CYK

Sam algorytm dość przystępnie jest umówiony tutaj: `https://www.youtube.com/watch?v=VTH1k-xiswM`

Przed rozpoczęciem wykonywania algorytmu musimy się więc upewnić, że nasza gramatyka jest w postaci CNF, a jeśli tak nie jest, to musimy wykonać konwersję do CNF.

Wniosek z wykonania algorytmu orzekamy na podstawie ostatniej wypełnianej komórki: wyraz należy do języka opisywanego przez gramatykę wtedy gdy symbol startowy należy do podzbioru zbioru, który wpisaliśmy w ostatnią wypełnianą kratkę, czyli na sam koniec powinniśmy dostać zbiór zawierający symbol startowy.

7 Uzupełnienie teoretyczne do egzaminu

W poprzednich rozdziałach znajdował się materiał teoretyczny stricte napisany pod ćwiczenia audytoryjne. Materiał ten w dużej mierze pokrywa się z tym co jest wymagane na egzamin, ale trochę wiedzy teoretycznej będzie konieczne do uzupełnienia. Ta wiedza uzupełniająca znajduje się w tym rozdziale. Polecam też zajrzeć pod ten link `http://zasoby.open.agh.edu.pl/~11sustrojny/index.html` - parę użytecznych informacji spisanych przez studenta dla studentów.

7.1. Gramatyka jednoznaczna i niejednoznaczna

Wiemy już z ćwiczeń numer 1, co to jest gramatyka, jak ją definiujemy, co opisuje. Teraz powiemy sobie co to znaczy, że gramatyka jest jednoznaczna lub niejednoznaczna, ale zanim do tego dojdziemy musimy wprowadzić kilka podstawowych pojęć.

7.1.1. Drzewo syntaktyczne (parse tree, drzewo parsowania)

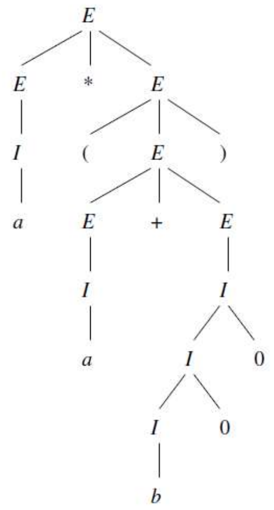
Drzewo syntaktyczne to po prostu graficzna reprezentacja wyprowadzenia danego słowa na podstawie produkcji dostępnych w danej gramatyce. Drzewo syntaktyczna tworzymy więc dla danej gramatyki $G = (V, T, P, S)$.

- w korzeniu drzewa znajduje się symbol startowy gramatyki, czyli aksjomat S
- na kolejnych poziomach, w każdym węźle umieszczamy pojedynczy symbol (terminalny lub nieterminalny) uzyskany w wyniku zastosowania wybranej produkcji

Przykładowo, mamy daną gramatykę, która zawiera następujące produkcje, gdzie symbolem startowym jest symbol E:

$$\begin{array}{lllll}
1. & E & \rightarrow & I \\
2. & E & \rightarrow & E+E \\
3. & E & \rightarrow & E*E \\
4. & E & \rightarrow & (E) \\
\\
5. & I & \rightarrow & a \\
6. & I & \rightarrow & b \\
7. & I & \rightarrow & Ia \\
8. & I & \rightarrow & Ib \\
9. & I & \rightarrow & I0 \\
10. & I & \rightarrow & I1
\end{array}$$

Przykładowe drzewo syntaktyczne dla tej gramatyki to:



Drzewo to reprezentuje wyprowadzenie słowa $a * (a + b00)$, co odpowiada zastosowaniu produkcji:

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow}$$
$$E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0) \underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow}$$
$$E * (I + b00) \underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

7.1.2. Wyprowadzenie lewostronne i prawostronne

Mając dane wyrażenie dla którego stosujemy produkcję często mamy wiele wyborów względem tego do którego symbolu zastosować produkcję. Czasem wygodnie jest nam przyjąć, którą ze zmiennych będziemy zamieniać w każdej kolejnej produkcji.
Wyprowadzenie w których w każdym kroku stosujemy produkcję do zmiennej znajdującej się najbardziej na lewo wyrażenia nazywamy wyprowadzeniami lewostronnymi. Jeśli taką samą regułę zastosujemy dla prawej strony, to wyprowadzenia takie nazywamy prawostronnymi.
Przykład dla gramatyki dla której produkcje opisaliśmy powyżej:

- wyprowadzenie lewostronne

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E \underset{lm}{\Rightarrow}$$
$$a * (E) \underset{lm}{\Rightarrow} a * (E + E) \underset{lm}{\Rightarrow} a * (I + E) \underset{lm}{\Rightarrow} a * (a + E) \underset{lm}{\Rightarrow}$$
$$a * (a + I) \underset{lm}{\Rightarrow} a * (a + I0) \underset{lm}{\Rightarrow} a * (a + I00) \underset{lm}{\Rightarrow} a * (a + b00)$$

- wyprowadzenie prawostronne

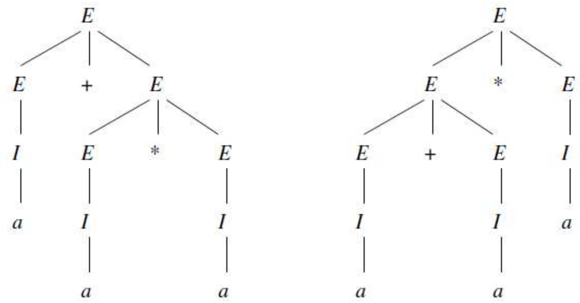
$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow}$$

$$E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0) \underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow}$$

$$E * (I + b00) \underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

7.1.3. Definicje

Zauważmy, że dla danej gramatyki możemy zastosować różna kolejność produkcji i uzyskać to samo słowo, przykładowo:



W tym przypadku otrzymujemy słowo $a + a * a$ stosując różną kolejność produkcji. Taką sytuację nazywamy niejednoznacznością gramatyki, tzn. możemy otrzymać to samo słowo stosując różną kolejność produkcji, czyli istnieje słowo dla którego można utworzyć dwa różne drzewa syntaktyczne.

Definicja - Gramatyka niejednoznaczna (wieloznaczna)

Mówimy, że gramatyka jest niejednoznaczna (wieloznaczna), gdy język opisywany przez tą gramatykę zawiera przynajmniej jedno zdanie niejednoznaczne (wieloznaczne). Gramatyka nie zawierająca zdania wieloznacznego to gramatyka jednoznaczna.

Definicja - Gramatyka jednoznaczna

Mówimy, że gramatyka jest jednoznacza jeżeli dla każdego zdania z języka opisywanego przez tą gramatykę istnieje dokładnie jedno drzewo syntaktyczne reprezentujące wyprowadzenie tego zdania, czyli gramatyka nie zawiera zdania wieloznacznego.

Definicja - Zdanie niejednoznaczne (wieloznaczne)

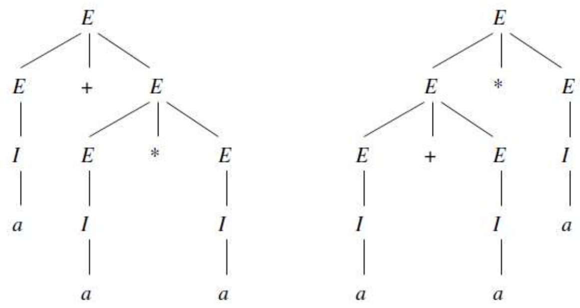
Mówimy, że zdanie jest wieloznaczne jeśli istnieje dla niego więcej niż jedno drzewo syntaktyczne wyprowadzenia takiego zdania, co jest równoważne temu, że istnieje więcej niż jedno lewostronne wyprowadzenie dla tego zdania, co jest równoważne temu, że istnieje więcej niż jedno prawostronne wyprowadzenie dla tego słowa. (Uzasadnienie lewostronnego i prawostronnego -> patrz następny podpodrozdział)

7.1.4. Eliminowanie niejednoznaczności

Nie ma algorytmu, który pozwoliłby wyeliminować niejednoznaczność z danej gramatyki, nie ma nawet algorytmu, który umiałby stwierdzić czy dana gramatyka jest niejednoznaczna czy jednoznaczna.

7.1.5. Sprawdzanie jednoznaczności

Można pokazać (Hopcroft), że w gramatykach jednoznacznych wszystkie wyprowadzenia lewostronne są różne pomiędzy sobą (prowadzą do różnych słów), a także wszystkie wyprowadzenia prawostronne są różne pomiędzy sobą. W przypadku gdy nasza gramatyka będzie posiadała dwa różne drzewa dla tego samego słowa, to na pewno znajdziemy je sprawdzając wyprowadzenia lewostronne lub prawostronne.
Dla rozważanego przez nas przykładu sprawdzamy wyprowadzenia lewostronne i otrzymujemy:



Co świadczy o tym, że gramatyka jest niejednoznaczna, ponieważ mamy tu dwa różne drzewa dla tego samego słowa.

7.2. Rozbiór gramatyczny + pojęcia i przykłady

Metody rozbioru gramatycznego (są dwie) to:

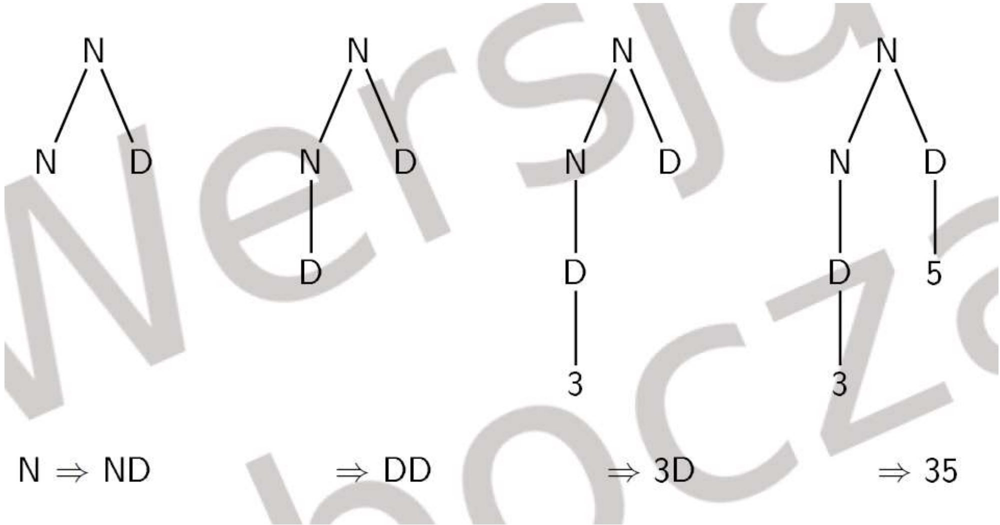
- metoda generacyjna - analiza rozpoczynana jest od symbolu wyróżnionego gramatyki, stosując kolejne produkcje do formuł zdaniowych aż do otrzymania wyłącznie symboli terminalnych (zdania)
- metoda redukcyjna (wstępująca) - analiza rozpoczyna się od zdania, stosując kolejne redukcje analizowanej formy zdaniowej dochodzimy do symbolu wyróżnionego gramatyki

Przykłady (ze starych wykładów Klimka)
Niech dana będzie gramatyka

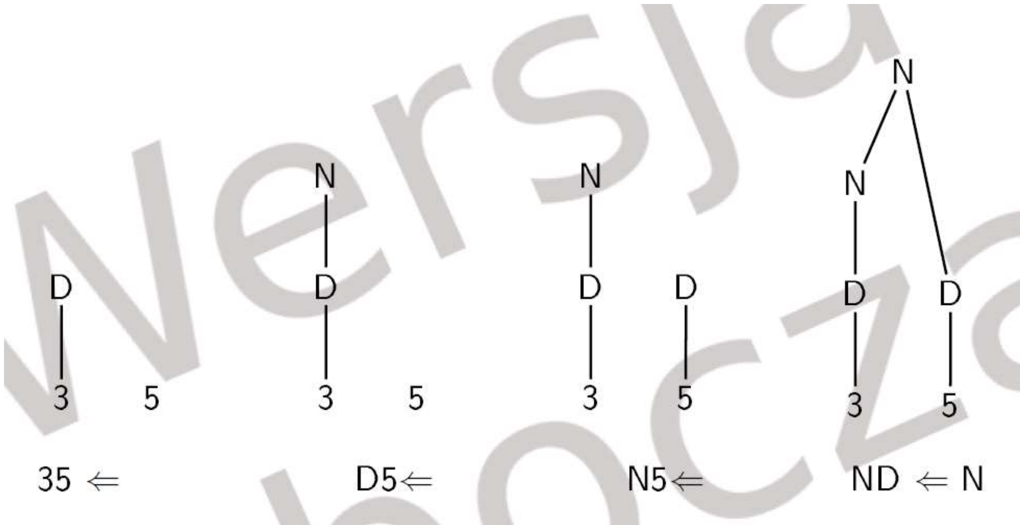
$$
\begin{aligned}
&G =< N, V, P, S > \\
&N = \{N, D\} \\
&V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\
&P = \{ \\
&N \to ND|D \\
&D \to 0|1|2|3|4|5|6|7|8|9\} \qquad\qquad\qquad S = N
\end{aligned}
$$

Dla zdania 35:

- metoda generacyjna:



- metoda zstępująca:



Definicja - Forma zdaniowa

To dowolny ciąg symboli terminalnych i nieterminalnych, ale wyprowadzalny z symbolu startowego.

Definicja - Zdanie

Zdanie to forma zdaniowa zawierająca tylko symbole terminalne.

Definicja - Rozbiór kanoniczny

Rozbiorem kanonicznym nazywamy rozbiór,który zawsze rozpoczyna się od lewej strony redukowanej formy zdaniowej,tzn. w pierwszej kolejności redukowane są lewostronne symbole formy.

Definicja - Wyprowadzenie

- Wyprowadzenie bezpośrednie - dla dowolnych $v, w \in (N \cup V)^*$, mówimy, że v wyprowadza w, co zapiszemy $v => w$ jeżeli istnieją takie $x, y \in (N \cup V)^*$ oraz produkcja $\alpha \rightarrow \beta$, że $v = x\alpha y$ oraz $w = x\beta y$
- Wyprowadzenie pośrednie - $v \stackrel{+}{\Longrightarrow} w$ jeśli istnieje ciąg napisów $u_0 u_1 \dots u_n, n > 0$ taki że $v = u_0 \Longrightarrow \cdots \Longrightarrow u_n = w$

Definicja - Uchwyt (osnowa, podstawa)

Uchwyt to podciąg symboli, który jest redukowany podczas rozbioru, mniej formalnie jest to wyrażenie, które znajduje się po prawej stronie produkcji, a które zamieniamy na wyrażenie po jej lewej stronie.

8 Egzamin - typy zadań + przykłady z lat poprzednich

8.1. Zadania teoretyczne ☑

8.1.1. Metody rozbioru gramatycznego ☑

> 2017/18, II
> 2015/16, I
> Podać metody rozbioru gramatycznego i krótko opisać zasady. Podać przykłady dla każdej z metod.

Patrz: Uzupełnienie teoretyczne do egzaminu.

8.1.2. Pojęcia ☑

> 2017/18, II
> Proszę wyjaśnić następujące pojęcia: forma zdaniowa, zdanie, rozbiór kanoniczny, wyprowadzenie, uchwyt.

Patrz uzupełnienie teoretyczne do egzaminu.

8.1.3. Definicja jednoznaczności ☑

> Wiki
> Jaka gramatyka nazywana jest jednoznaczną, a jaka wieloznaczną ? Podaj przykład gramatyki wieloznacznej i dwóch różnych wyprowadzeń dla tego samego słowa.

2017/18, I
Brak konkretnego polecenia dla tego zadania na wiki / poprzednich egzaminach.
Patrz: Uzupełnienie teoretyczne do egzaminu.

8.1.4. Definicja stanów nierozróżnialnych + przykład ☑

> 2013/14, I
> Definicja stanów nierozróżnialnych + przykład

Patrz równoważność i minimalizacja automatów. Przykład dość łatwo wymyślić rysując jakiś prosty 'symetryczny' automat czy też przejścia od dwóch stanów wchodzące do jednego. Rozróżnialne, to wystarczy podać przykładowo jeden akceptujący, a drugi nieakceptujący.

UWAGA: Lepsza będzie bardziej formalna, krótsza definicja:
Dwa stany p oraz q są równoważne jeśli

For all input strings w, $\hat{\delta}(p, w)$ is an accepting state if and only if $\hat{\delta}(q, w)$ is an accepting state.

8.1.5. Klasyfikacja Chomsky'ego i jakie ma odniesienie do automatów ☑

2013/14, II Klasyfikacja Chomsky'ego i jakie ma odniesienie do automatów

Patrz Hierarchia Chomsky'ego.
Odniesienie do automatów: automaty reprezentują tylko języki regularne, czyli języki klasy 3 z hierarchii Chomsky'ego.

8.2. Zadania praktyczne

8.2.1. Skonstruować gramatykę dla... ☑

Ogólnie w tego typu zadaniach nie mamy jakiegoś algorytmu konstrukcji gramatyki. Musimy kierować się tym jak wyglądają słowa i na tej podstawie tworzyć gramatykę. Pamiętajmy, że gramatyka to cała czwórka więc musimy opisać każdą jej składową.

Spis wszystkich języków z poniższych zadań:

- $L(G) = \{a^m : m = 3n, n \geq 1\}$
- $L = \{s : s = \nabla^{2^i}, i \geq 1\}$
- $a^n b^{n+m} a^{m+k} b^k$
- $(a^n b^{n+k} a^k)$
- $a^n b^n c^n$
- ...
- $a^n b^{2n} c^n, n > 0$
- $a^n b^{2n} c^{3n}, n > 0$
- $a, a2, a4, a8, \ldots$
- $a^n b^m c^m d^{2n}, n \geq 0, m > 0$
- $a^n b^m, 0 \leq n \leq m \leq 2n$
- $a^n b^m c^k, k = n + m$
- $a^n b^m c^k, k \neq n + m$
- $a^n b^m c^k, n \neq m \vee m \neq k$

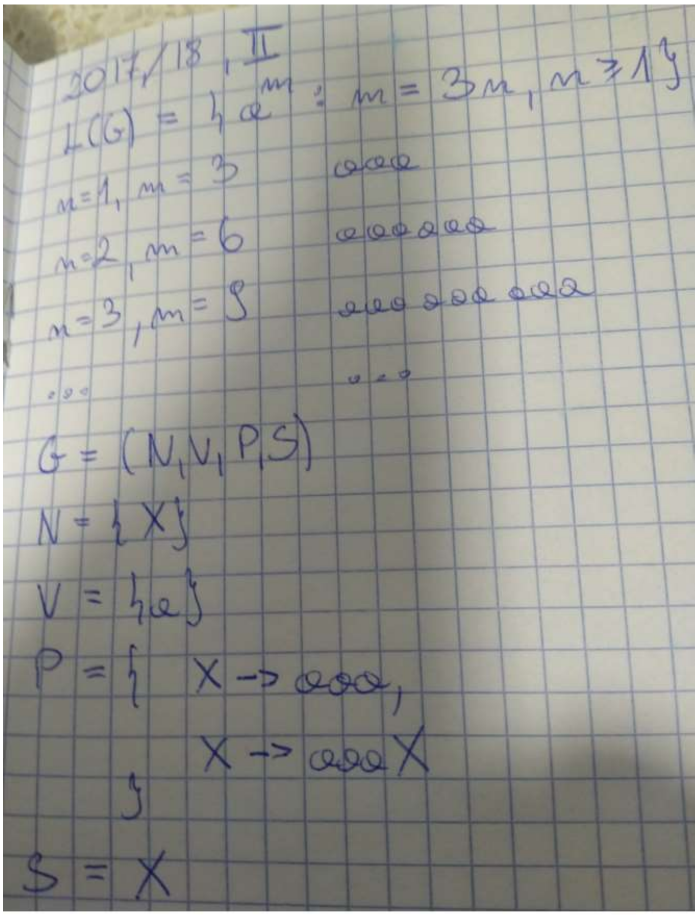
2017/18, II
Zaproponuj gramatykę klasy 0 lub 1 dla języka

$$L(G) = \{a^m : m = 3n, n \geq 1\}$$

2016/17, I
Proszę skonstruować gramatykę generującą język:

$$L = \{s : s = a^m, m = 3n, n \geq 1\}$$

Tabelka, wiki
Gramatyka dla a^{3n}


2017/18, II
$L(G) = \{a^m : m = 3n, n \geq 1\}$
$n=1, m=3$ aaa
$n=2, m=6$ aaaaaa
$n=3, m=9$ aaa aaa aaa
... ...
$G = (N, V, P, S)$
$N = \{X\}$
$V = \{a\}$
$P = \{ X \to aaa,$
$X \to aaaX$
$\}$
$S = X$

UWAGA: Zależy jakie jest ograniczenie, dla ograniczenia większe bądź równe 0 konieczne byłby dodanie produkcji z ε po prawej stronie. $X \to \varepsilon$, $X \to aaaX$.

2017/18, II
Skonstruować gramatykę generującą te i tylko te słowa które należą do języka

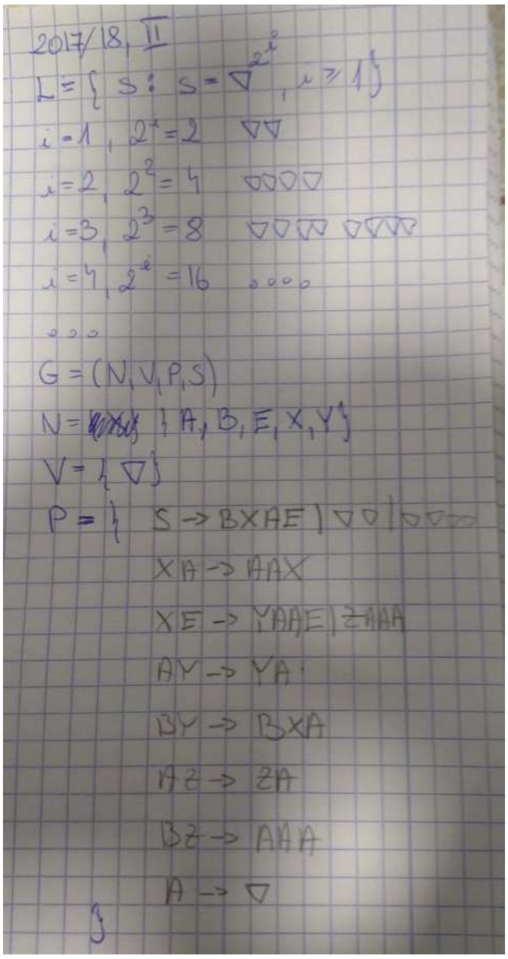
$$L = \{s : s = \nabla^{2^i}, i \geq 1\}$$

Tabelka, wiki
Gramatyka dla a^{2^i}

Rozwiązanie z dokumentu:

$\boldsymbol{a^{2^i}, i \geq 0}$

$N = \{L, Q, X\}$
$V = \{a\}$
$P = \{X \to a, X \to LXQ, La \to aaL, LLQQ \to LQ, aLQ \to a\}$
$S = X$



2017/18, II

$L = \{ s : s = \nabla^{2^i}, i \geq 1 \}$

$i = 1,\ 2^1 = 2$ $\nabla\nabla$

$i = 2,\ 2^2 = 4$ $\nabla\nabla\nabla\nabla$

$i = 3,\ 2^3 = 8$ $\nabla\nabla\nabla\nabla\ \nabla\nabla\nabla\nabla$

$i = 4,\ 2^4 = 16$ ....

...

$G = (N, V, P, S)$

$N = \{A, B, E, X, Y\}$

$V = \{\nabla\}$

$P = \{$ $S \to BXAE \mid \nabla\nabla \mid \nabla\nabla\nabla\nabla$

$XA \to AAX$

$XE \to YAAE \mid ZAAA$

$AY \to YA$

$BY \to BXA$

$AZ \to ZA$

$BZ \to AAA$

$A \to \nabla$

$\}$

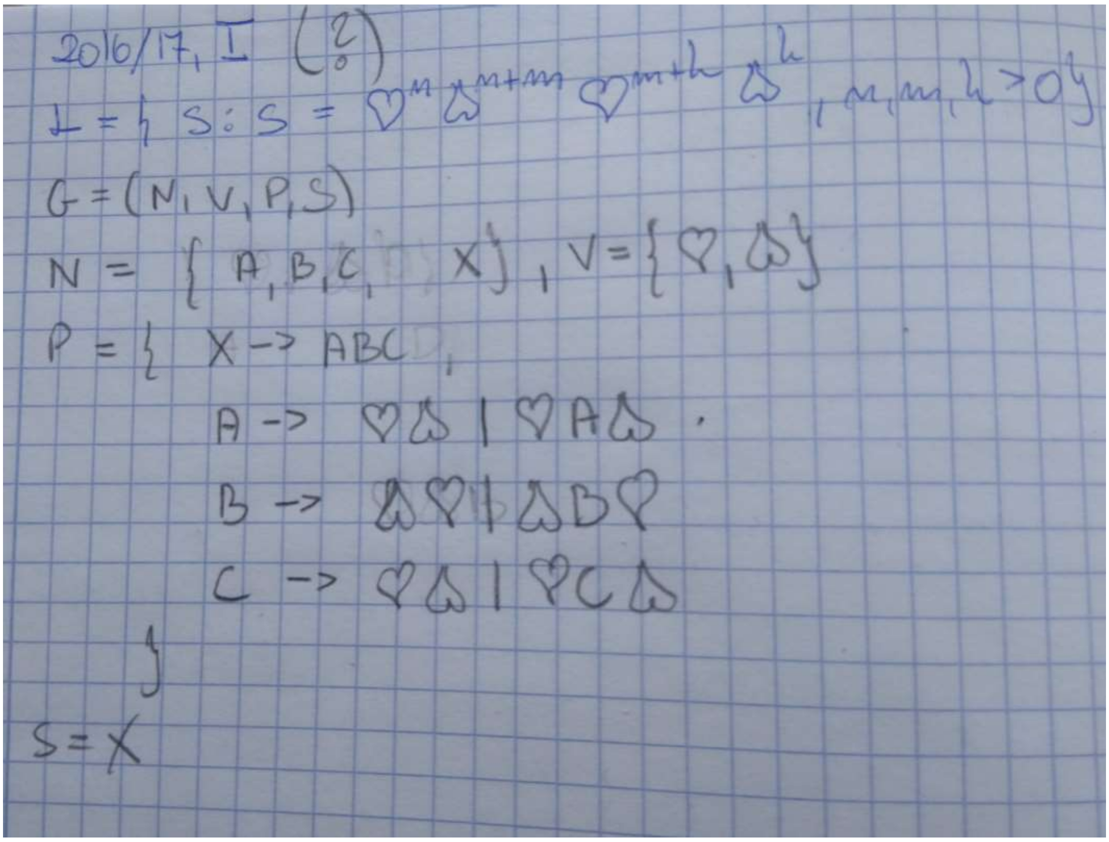
2016/17, I

Proszę skonstruować gramatykę generującą język:

$$L = \{ s : s = \heartsuit^n \spadesuit^{n+m} \heartsuit^{m+k} \spadesuit^k \quad n, m, k > 0 \}$$

Tabelka, wiki

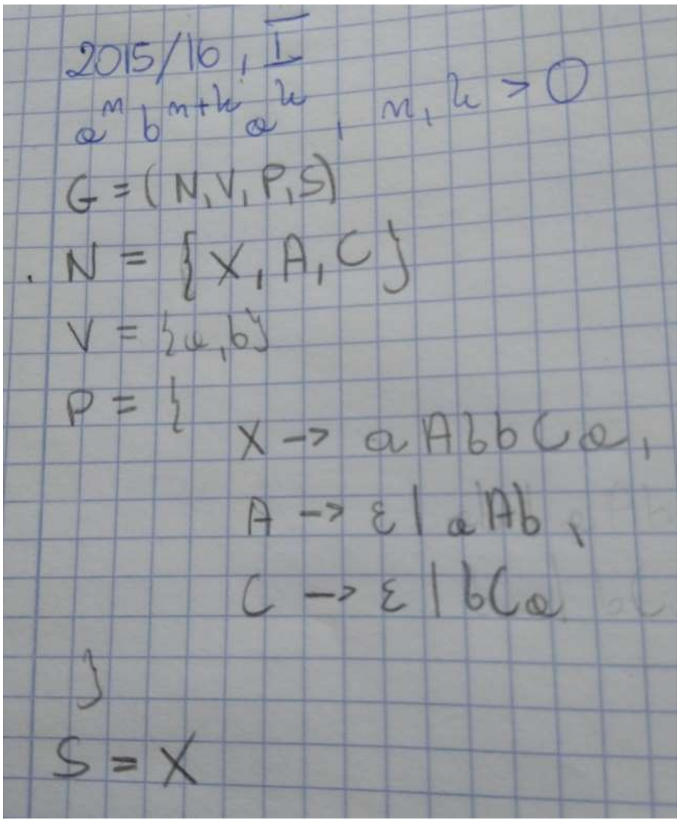
Gramatyka dla $a^n b^{n+m} a^{m+k} b^k$



2015/16, I
Jaką gramatykę generuje

$$(a^n b^{n+k} a^k)$$

Tabelka, wiki
Gramatyka dla $a^n b^{n+k} a^k$



2015/16, I

$a^n b^{n+k} a^k$, $n, k > 0$

$G = (N, V, P, S)$

$N = \{X, A, C\}$

$V = \{a, b\}$

$P = \{$

$X \to aAbbCa$,

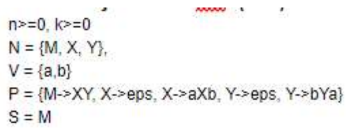
$A \to \varepsilon \mid aAb$,

$C \to \varepsilon \mid bCa$

$\}$

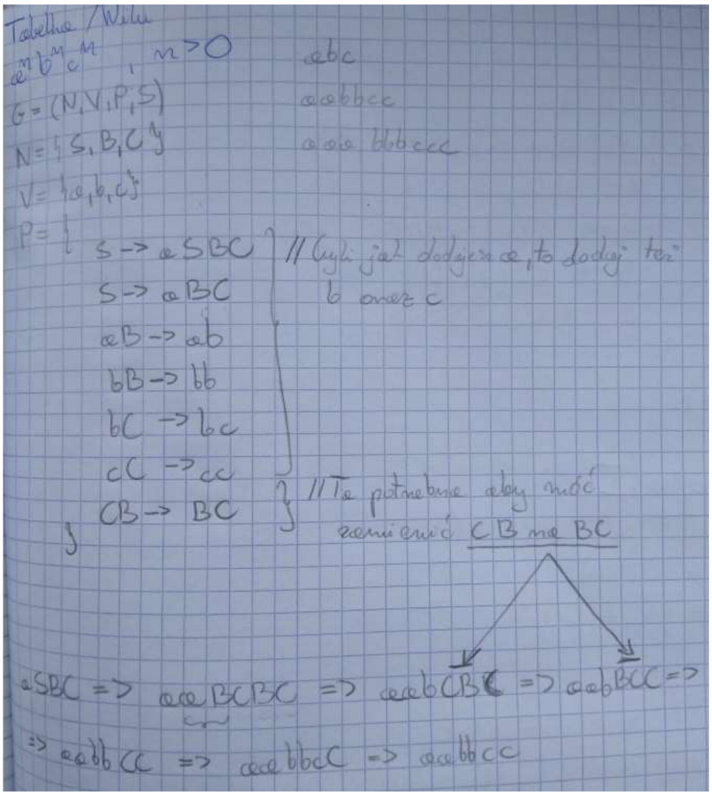
$S = X$

Chyba dla innych ograniczeń:



n>=0, k>=0
N = {M, X, Y},
V = {a,b}
P = {M->XY, X->eps, X->aXb, Y->eps, Y->bYa}
S = M

Tabelka, wiki
Gramatyka dla $a^n b^n c^n$



Tabelka / Wiki

$a^n b^n c^n$, $n > 0$

abc

aabbcc

aaabbbccc

$G = (N, V, P, S)$

$N = \{S, B, C\}$

$V = \{a, b, c\}$

$P = \{$

$S \to aSBC$

$S \to aBC$

$aB \to ab$

$bB \to bb$

$bC \to bc$

$cC \to cc$

$CB \to BC$

$\}$

// Czyli jak dodajesz a, to dodaj też b oraz c

// Te potrzebne aby móc zamienić CB na BC

$aSBC \Rightarrow aaBCBC \Rightarrow aabCBC \Rightarrow aabBCC \Rightarrow$

$\Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcc$

Jakoś inaczej z dokumentu:

$\boldsymbol{a^n b^n c^n}, \qquad \boldsymbol{n} > 0$

$N = \{A, B, L\}$

$V = \{a, b, c\}$

$P = \{\, B \to ABc, A \to aL, aLa \to aaL, LB \to Bb, aLB \to b \,\}$

$S = B$

Opracowanie zadań

$a^n b^{2n} c^n, n > 0$

Opracowanie zadań

$a^n b^{2n} c^{3n}, n > 0$

Opracowanie zadań

$a, a2, a4, a8, \ldots$

Opracowanie zadań
$a^n b^m c^m d^{2n}, n \geq 0, m > 0$

Opracowanie zadań
$a^n b^m, 0 \leq n \leq m \leq 2n$

Opracowanie zadań
$a^n b^m c^k, k = n + m$

Opracowanie zadań
$a^n b^m c^k, k \neq n + m$

Opracowanie zadań
$a^n b^m c^k, n \neq m \vee m \neq k$

Gramatyka dla liczb ze znakiem lub bez, podzielnych przez 5

$\boldsymbol{a^n b^n c^n}, \quad \boldsymbol{n} > 0$

$N = \{A, B, L\}$
$V = \{a, b, c\}$
$P = \{ B \rightarrow ABc, A \rightarrow aL, aLa \rightarrow aaL, LB \rightarrow Bb, aLB \rightarrow b \}$
$S = B$

```
S -> Z
Z -> +L|-L|L
 L -> WX|X
X -> 0|5
W -> 1|2|3|4|5|6|7|8|9|0|WW
```

8.2.2. Jaki język generuje gramatyka... $\boxslash$

W tego typu zadaniu również trudno o jakiś konkretny algorytm według którego postępujemy. Dobrze jest rozpisać sobie słowa należące do języka. Na koniec język można opisać przy pomocy zbioru słów, przykładowo $J(G) = \{zy(xy)^n z, n \geq 0\}$

2017/18, II
Jaki język generuje gramatyka $G = (N, V, P, S)$

$$\begin{aligned} &N = \{S, X, L, P\} \\ &V = \{[,]\} \\ &P = \{S \to LX|LP|SS, X \to SP, L \to [, P \to]\} \\ &S = S \end{aligned}$$

2016/17, I
Jaki język generuje gramatyka ?

$$\begin{aligned} &S -> LX|LP|SS \\ &X -> SP \\ &L -> [\\ &P ->] \end{aligned}$$

Można rozpisać parę przykładów...
Poprawne nawiasowanie - gramatyka ta generuje język do którego należą słowa składające się z symboli [oraz] takie, że:

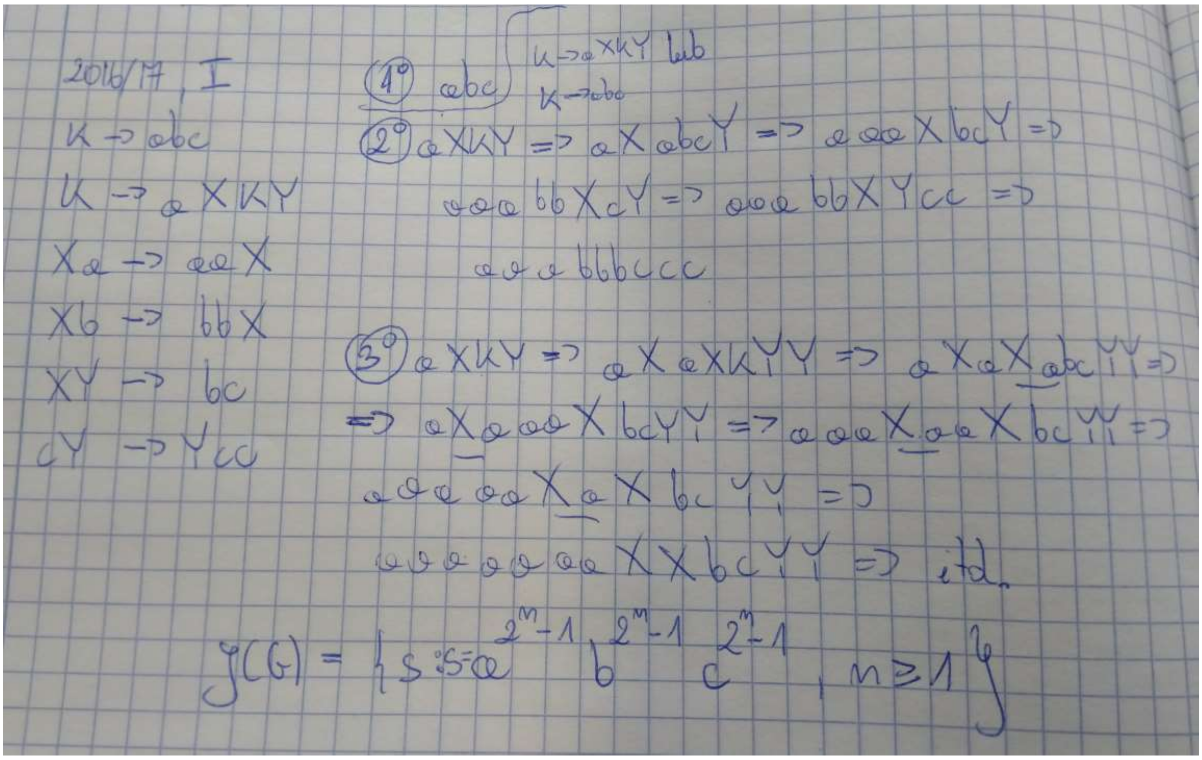
- W każdym słowie występuje tyle samo symboli [co symboli]
- Każdy prefiks słowa zawiera co najmniej tyle symboli [(nawiasów otwierających), co symboli] (nawiasów zamykających)

2016/17, I
Jaki język generuje gramatyka ?

$$\begin{aligned} K &-> abc \\ K &-> aXKY \\ Xa &-> aaX \\ Xb &-> bbX \\ XY &-> bc \\ cY &-> Ycc \end{aligned}$$

Tabelka, wiki
Jaki język generuje gramatyka... podana gramatyka taka, że odpowiedź to:

$$a^{(2^n)-1}b^{(2^n)-1}c^{(2^n)-1}$$



Tabelka, wiki
Jaki język generuje gramatyka ?

$$R \to LRQ$$
$$La \to aaL$$
$$bQ \to Qbb$$
$$LLQQ \to LQ$$
$$aLQb \to ab$$

Zadanie 1.5

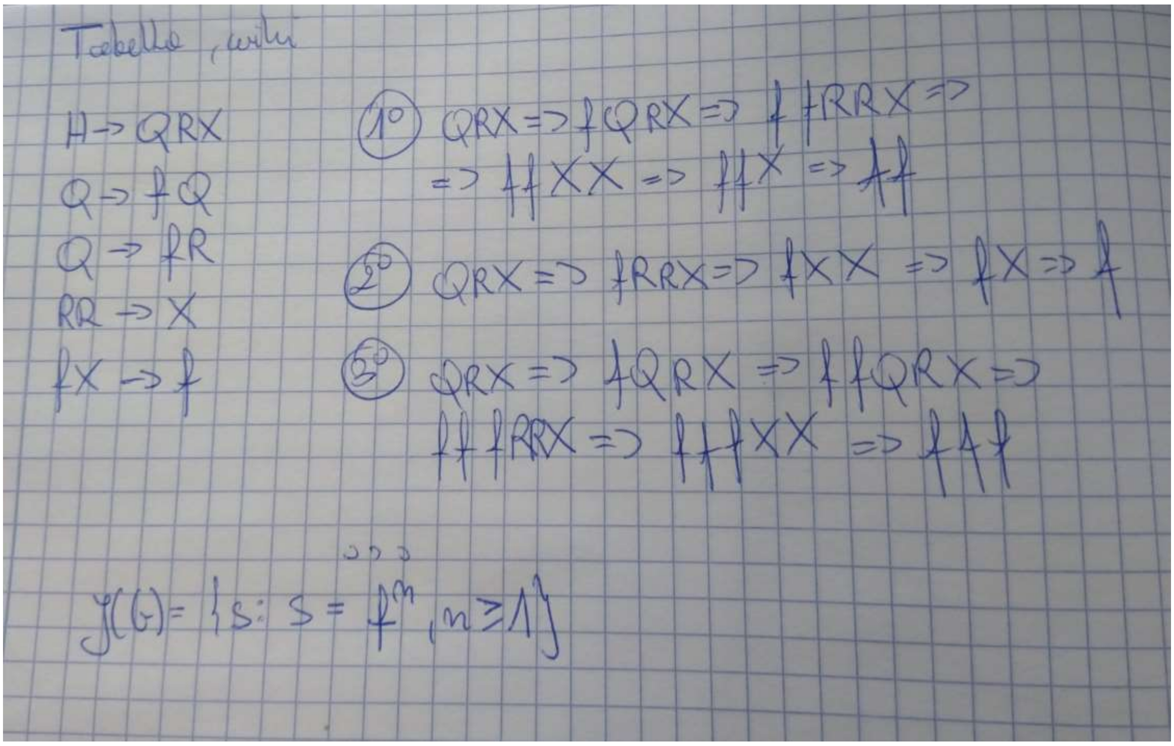
Jaki język jest generowany przez gramatykę $G = \langle N, V, P, S \rangle$:
$N = \{L, Q, R\}$
$V = \{a, b\}$
$P = \{R \to ab, R \to LRQ, La \to aaL, bQ \to Qbb, LLQQ \to LQ, aLQb \to ab\}$
$S = R$

Odpowiedź

Gramatyka generuje język zawierający słowa postaci $a^{2^n} b^{2^n}$, gdzie $n \geq 0$.

Tabelka, wiki
Jaki język generuje gramatyka ?

$$H \to QRX$$
$$Q \to fQ$$
$$Q \to fR$$
$$RR \to X$$
$$fX \to f$$


Tabelka, wiki

H -> QRX
Q -> fQ
Q -> fR
RR -> X
fX -> f

1° QRX => fQRX => ffRRX => ffXX => ffX => ff
2° QRX => fRRX => fXX => fX => f
3° QRX => fQRX => ffQRX => fffRRX => fffXX => fff

$J(G) = \{s : s = f^n, n \geq 1\}$

Zadanie 1.6

Jaki język jest generowany przez gramatykę $G = \langle N, V, P, S \rangle$:
$N = \{K, X, Y\}$
$V = \{a, b, c\}$
$P = \{K \to abc, K \to aXKY, Xa \to aaX, Xb \to bbX, XY \to bc, cY \to Ycc\}$
$S = K$

Odpowiedź

Gramatyka generuje język zawierający słowa postaci $a^{2^n-1}b^{2^n-1}c^{2^n-1}$, gdzie $n \geq 1$.

Zadanie 1.4

Jaki język jest generowany przez gramatykę $G = \langle N, V, P, S \rangle$:
$N = \{H, Q, R, X\}$
$V = \{f\}$
$P = \{H \to QRX, Q \to fQ, Q \to fR, RR \to X, fX \to f\}$
$S = H$
Czy istnieje gramatyka klasy wyższej generująca ten sam język?

Odpowiedź

Gramatyka generuje język zawierający słowa postaci f^n, gdzie $n \geq 1$. Gramatyka zawierająca produkcje $P = \{H \to f, H \to fH\}$ oraz symbol wyróżniony $S = H$ generuje ten sam język i jest to gramatyka regularna.

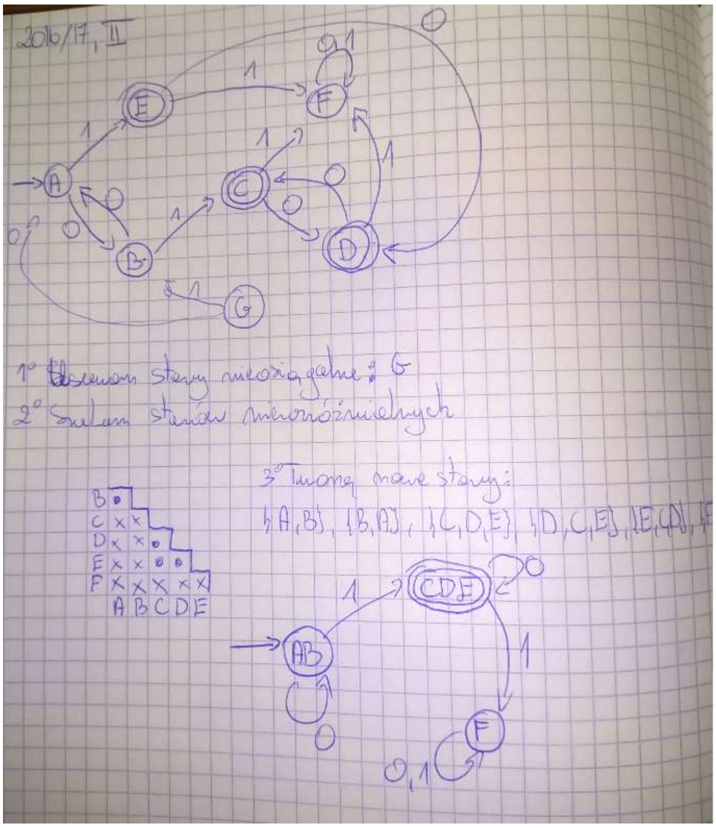
8.2.3. Minimalizacja automatu ☑

2017/18, II

Poddać minimalizacji pokazując kolejne kroki postępowania a także automat wynikowy w postaci graficznej następujący automat skończony

$$A = (\{A, B, C, D, E, F, G\}, \{0, 1\}, \delta, A, \{C, D, E\})$$

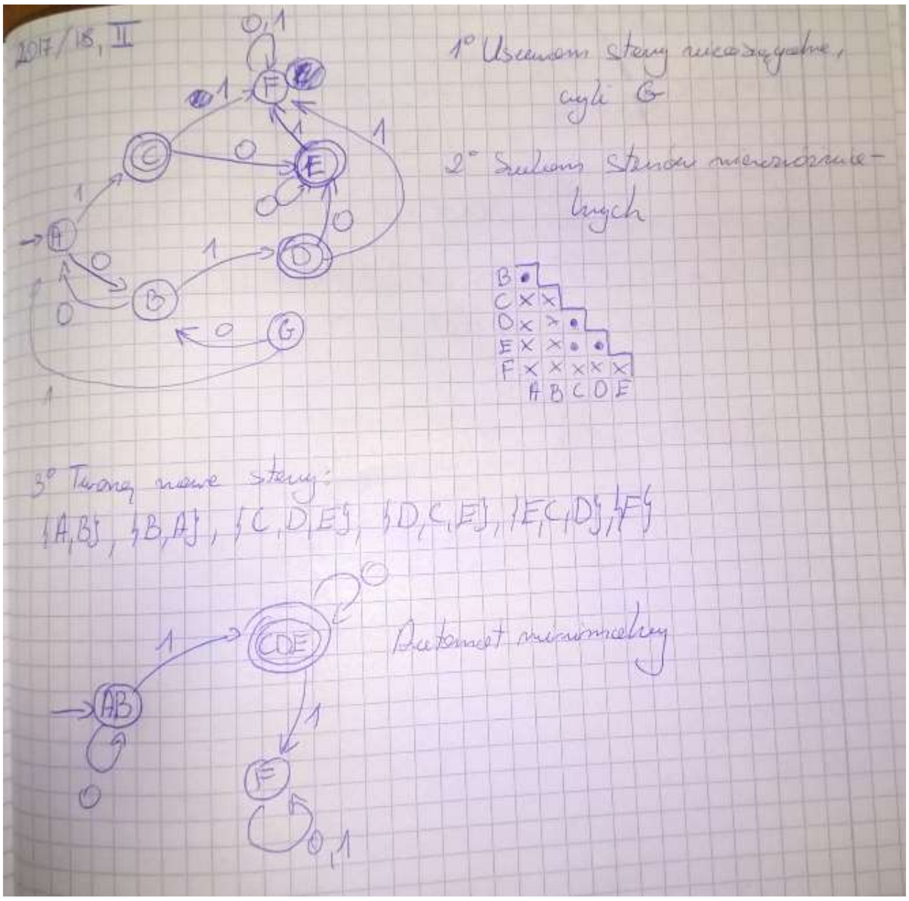
$$\begin{array}{ll}
\delta(A,0) = B & \delta(A,1) = C \\
\delta(B,0) = A & \delta(B,1) = D \\
\delta(C,0) = E & \delta(C,1) = F \\
\delta(D,0) = E & \delta(D,1) = F \\
\delta(E,0) = E & \delta(E,1) = F \\
\delta(F,0) = F & \delta(F,1) = F \\
\delta(G,0) = B & \delta(G,1) = A
\end{array}$$



2016/17, II

Proszę zminimalizować następujący automat:

$Automat = \{\{A, B, C, D, E, F, G\}, \{0, 1\}, \{\delta(A, 0) = B, \delta(A, 1) = E, \delta(B, 0) = A, \delta(B, 1) = C, \delta(C, 0) = D, \delta(C, 1) = F, \delta(D, 0) = C, \delta(D, 1) = F, \delta(E, 0) = D, \delta(E, 1) = F, \delta(F, 0) = F, \delta(F, 1) = F, \delta(G, 0) = A, \delta(G, 1) = B\}, A, \{C, D, E\}\}$



8.2.4. Sprawdzić stosując CYK, czy zdanie ... należy do podanej gramatyki ☑

Uwaga, w jednym z zadań było podane sprawdzenie dla 'aabbb' oraz 'aaabbb', w takim przypadku jak rozpiszemy dla 'aaabbb' i zasłonimy pierwsza kolumnę, to mamy rozwiązanie dla 'aabbb'.

Egzamin 2013/14, I

Zadanie.1 Sprawdzić `*[*#]#`

$$S- > S\#$$
$$S- > G$$
$$G- > *[S]$$
$$G- > *$$

Egzamin 2013/14

S -> S#

S -> G

G -> *[S]

G -> *

Wejście *[*#]#

Konwersja do CNF:

S -> S# | *[S] | *

G -> *[S] | *

S -> SH | MLSP | *

G -> MLSP | *

S -> SH | MN | *

G -> MN | *

N -> LA

A -> SP

M -> *

L -> [

P ->]

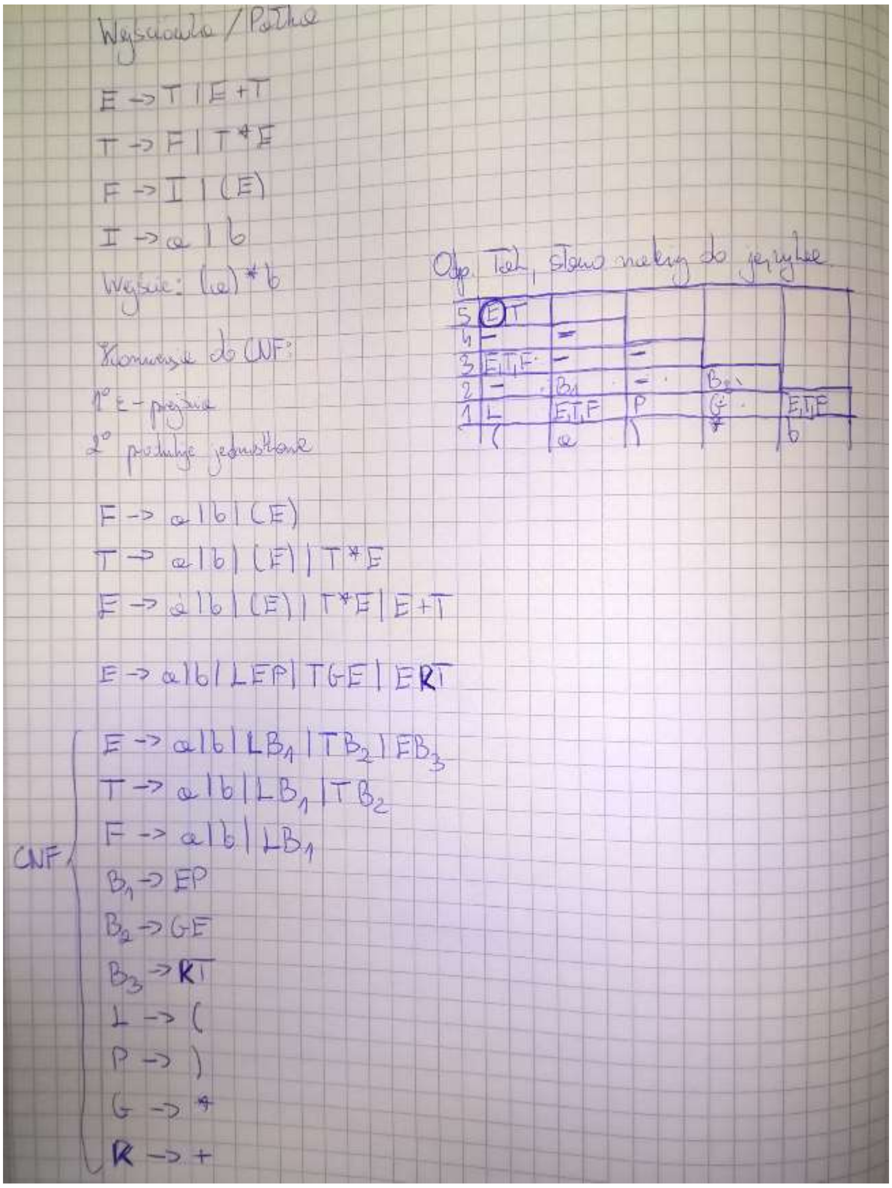
H -> #

} CNF

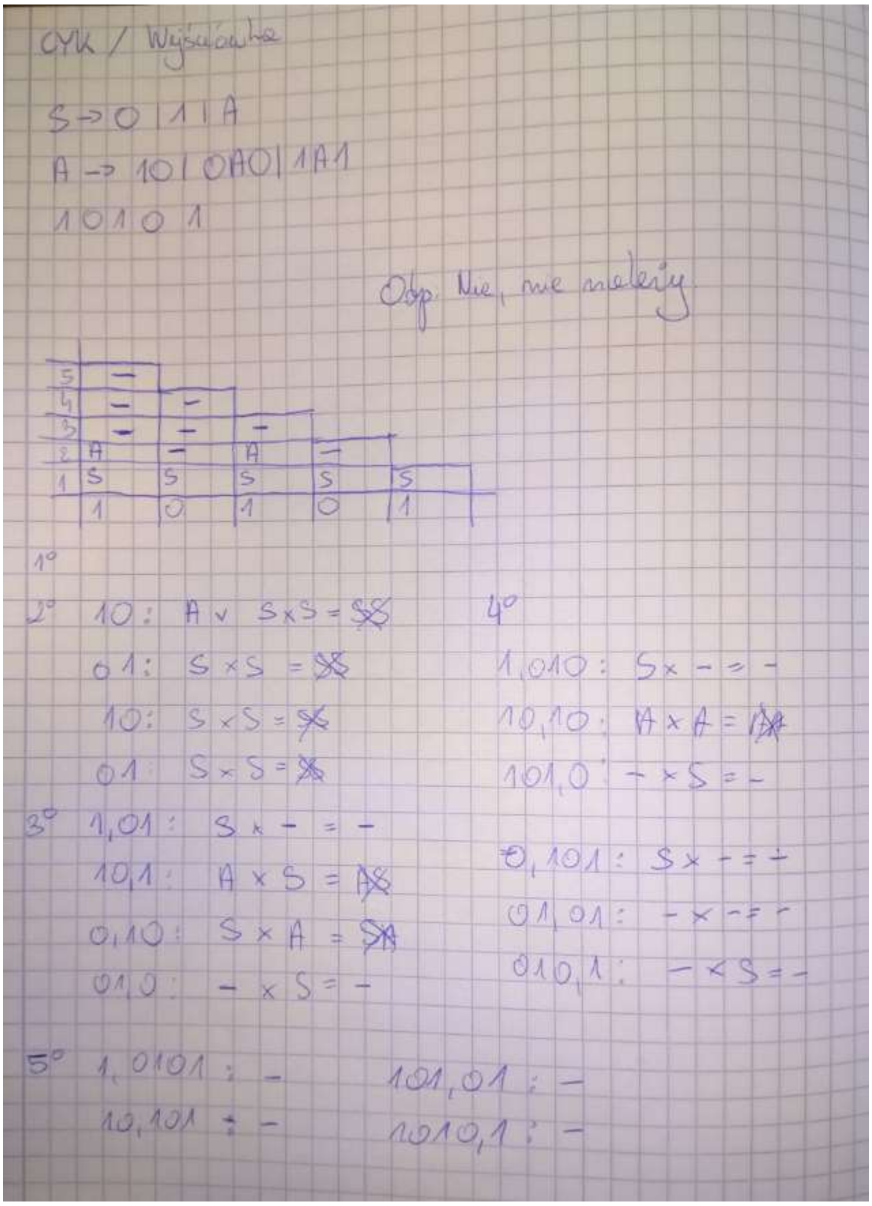
Odp: Tak, słowo należy do J(G)

6	S					
5	S,G	–				
4	–	N	–			
3	–	–	~~S~~A	–		
2	–	–	S	–	–	
1	S,G,M	L	S,G,M	H	P	H
	*	[	*	#	]	#

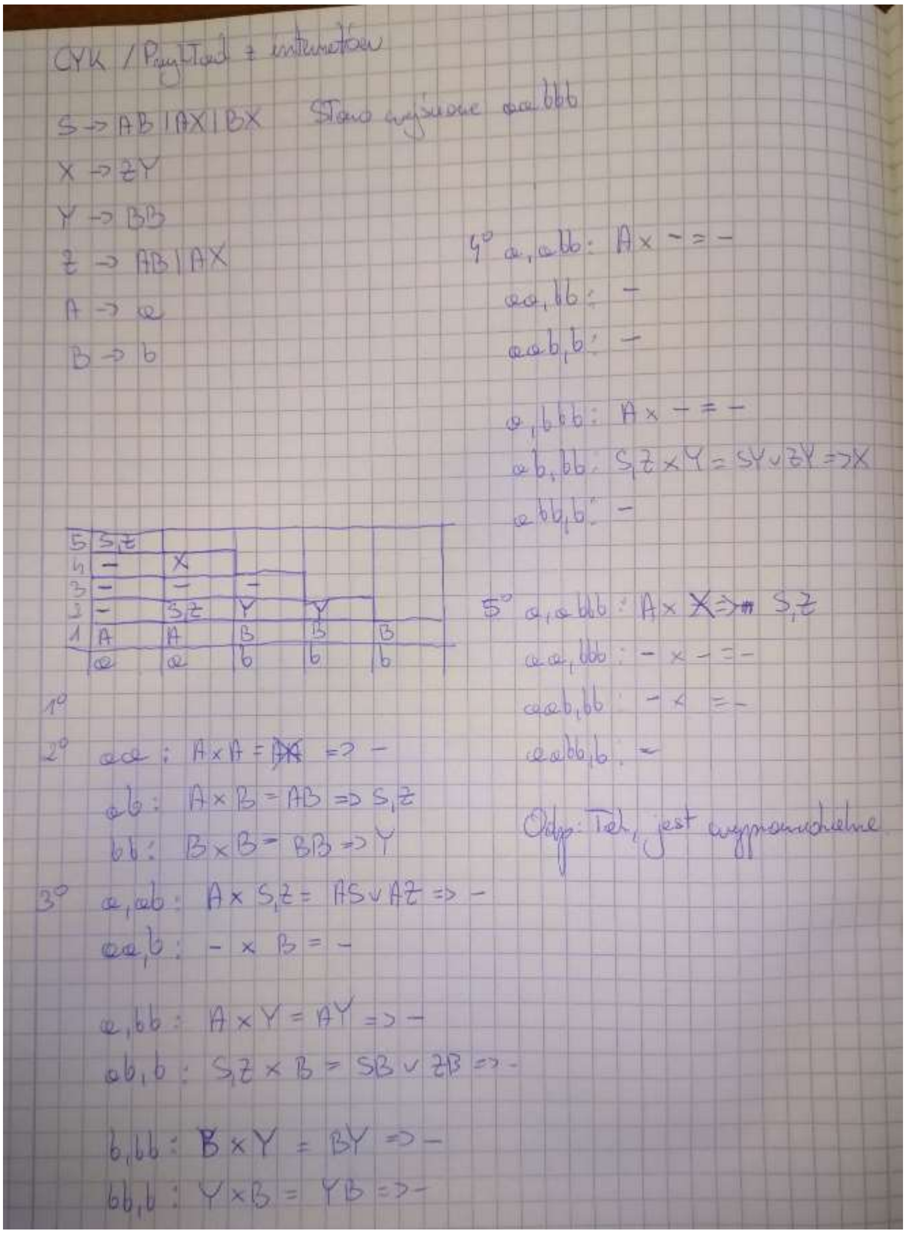
Przykład z wejściówki



Przykład z Hopcrofta (bez konwersji, mało prawdopodobne)



Przykład z internetów (bez konwersji, mało prawdopodobne)



CYK / Przykład z internetów

S -> AB | AX | BX   Słowo wejściowe aabbb

X -> ZY

Y -> BB

Z -> AB | AX

A -> a

B -> b

5	S,Z				
4	–	X			
3	–	–	–		
2	–	S,Z	Y	Y	
1	A	A	B	B	B
	a	a	b	b	b

1°

2° aa : A × A = ~~AA~~ => –

ab : A × B = AB => S,Z

bb : B × B = BB => Y

3° a,ab : A × S,Z = AS ∨ AZ => –

aa,b : – × B = –

a,bb : A × Y = AY => –

ab,b : S,Z × B = SB ∨ ZB => –

b,bb : B × Y = BY => –

bb,b : Y × B = YB => –

4° a,abb : A × – = –

aa,bb : –

aab,b : –

a,bbb : A × – = –

ab,bb : S,Z × Y = SY ∨ ZY => X

abb,b : –

5° a,abbb : A × X => S,Z

aa,bbb : – × – = –

aab,bb : – × – = –

aabb,b : –

Odp: Tak, jest wyprowadzalne

8.2.5. Typ gramatyki generującej $a^k b^l c^m$ w zależności od k,l,m ☑

Tabelka, wiki
Typ gramatyki generującej $a^k b^l c^m$ w zależności od k,l,m

2016/17, II
Dany jest język $L = \{a^k b^l c^m : k, l, m > 0\}$. Jakie warunki należałoby nałożyć na k, l, m aby język znalazł się w 3,2,1 klasie Chomskiego ? (Gdzie brak warunku również jest specyficznym warunkiem). Proszę zaproponować gramatykę generującą ten język w danych klasach.

- dla braku warunków (poza przyjętym $k, l, m > 0$) - język regularny Przykład takiej gramatyki:

$$\begin{aligned}
&S \to aA \\
&A \to aA|bB \\
&B \to bB|cC \\
&C \to cC|\varepsilon
\end{aligned}$$

- $k = l \lor l = m \lor k = m$ - język bezkontekstowy Przykładowo dla $l = k$, język jest postaci $a^k b^k l^m$, a gramatyka ma postać:

$$\begin{aligned}
&S \to aAbcM \\
&A \to ab|aAb|\varepsilon \\
&M \to cM|\varepsilon
\end{aligned}$$

Języki takiej postaci nie są regularne, co wynika z lematu o pompowaniu:

$$\begin{aligned}
&w = a^k b^k c^m \\
&w = xyz \\
&\text{Niech } xy = \underbrace{aaaa\ldots a}_{n} \\
&\text{Oraz } x = \underbrace{aaaa\ldots a}_{n-1} \\
&\text{Wtedy } z = \underbrace{bbb\ldots b}_{n} cccc\ldots c \\
&\text{Dla } k = 0: \\
&w = xy^k z = xy = \underbrace{a\ldots a}_{n-1}\underbrace{bbb\ldots b}_{n} ccc\ldots c \\
&\text{Nie należy do jezyka}
\end{aligned}$$

- dla $k = l = m$ język kontekstowy Język jest postaci $a^k b^k c^k$, przykładowa gramatyka:

$$
\begin{aligned}
&S \to aBC \\
&S \to aSBC \\
&CB \to CZ \\
&CZ \to WZ \\
&WZ \to WC \\
&aB \to ab \\
&bB \to bb \\
&bC \to bc \\
&cC \to cc
\end{aligned}
$$

Języki takiej postaci nie są bezkontekstowe, co wynika z lematu o pompowaniu:

$z = a^n b^n c^n = uvwxy$
$|vwx| \leq n \implies$ vwx nie moze zawierac zarowno a jak i c
$\implies$ vx sklada sie tylko z a i b
Niech $z = uv^i wx^i y,$ dla $i = 0$:
$z = uwy$, gdzie y zawiera n znakow c, a uw mniej nic n znakow a lub mniej niz n znakow b
$\implies$ nie nalezy do jezyka $\implies$ nie jest bezkontekstowy

8.2.6. Jaki język generuje G ? Czy jest to gramatyka jednoznaczna ? Jeśli nie, to podać przykład słowa dla którego istnieją dwa wywody.

2016/17, II

Jaki język generuje G ? Czy jest to gramatyka jednoznaczna ? Jeśli nie, to podać przykład słowa dla którego istnieją dwa wywody.

$G = \{N, V, P, S\}$
$N = \{A, B, C, D, E, F, G, H, I, J\}$
$V = \{f, u\}$
$S = F$
$P = \{F \rightarrow uf\ F \rightarrow FuGH, fH \rightarrow fuffG, uH \rightarrow uJ, fI \rightarrow If, fJ \rightarrow Jf$
i w ciul innych dziwnych produkcji które nie mają sensu, na końcu pojawiła się taka:
$ufJK \rightarrow f\}$
gdzie K nie było nigdzie wcześniej.
Klimka chyba poniosła fantazja na festiwalu spierdolenia