# Teoretyczne podstawy informatyki

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. letni 2025/26

inf.ug.edu.pl/~amb/TPI





# Gramatyki. Języki bezkontekstowe.





## Gramatyki, uzasadnienie

- liczby definiowaliśmy wyrażeniem (niedokładnie, nie ma samego zera) $znak\ cyfra\_nie0\ cyfra^*(\varepsilon \mid .\ cyfra\ cyfra^*)$ gdzie $znak = \varepsilon \mid + \mid -$, $cyfra\_nie0 = 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$, $cyfra = 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ używając krótszych oznaczeń $ZC_0C^*(\varepsilon \mid .CC^*)$ gdzie $Z = \varepsilon \mid + \mid -$, $C_0 = 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$, $C = 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
- równości definiowały podstawienie, które dało wyrażenie dla liczb
- można pisać również $C = 0 \mid C_0$ i wykonać dwa podstawienia
- można też zrezygnować z definiowania liczby podstawień, zostawić tyle ile potrzeba dla danego słowa, pozwala to zrezygnować z operacji gwiazdki
- słowa pasujące do wzorca $C^*$ pasują do $S = \varepsilon \mid CS$





## Gramatyki, uzasadnienie c.d.

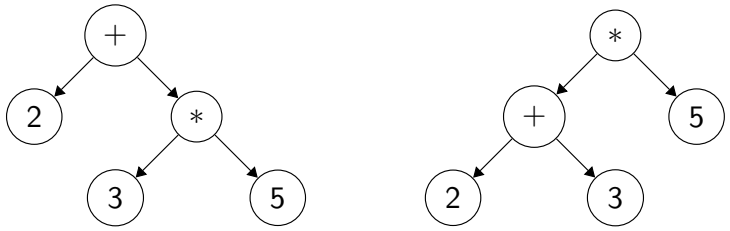
- definiujemy liczby następująco

| | | |
|---|---|---|
| $L$ | $\to IF$ | część całkowita i ułamkowa |
| $I$ | $\to ZC_0S$ | znak, cyfra niezerowa, ciąg cyfr dowolny |
| $Z$ | $\to \varepsilon \mid + \mid -$ | |
| $C_0$ | $\to 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ | |
| $S$ | $\to \varepsilon \mid CS$ | zastępuje operator Kleene'ego |
| $C$ | $\to 0 \mid C_0$ | |
| $F$ | $\to \varepsilon \mid .CS$ | kropka, cyfra, ciąg cyfr |

- ostateczne słowo ma pasować do $L$ i być słowem nad alfabetem $T = \{+, -, ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- symbole $V = \{L, I, F, Z, C_0, S, C\}$ są jedynie pomocnicze, $L$ jest symbolem początkowym, od którego rozpoczynamy próbę dopasowania słowa
- każdy z powyższych wierszy definiuje podstawienie $V \to \mathcal{P}((V \cup T)^*)$ a dokładniej, podzbiory będą skończone i podane enumeratywnie

## Gramatyki

- Wyrażenie $\to$ Liczba | Wyrażenie Operator Wyrażenie
  Operator $\to + \mid - \mid * \mid \div$
  Liczba (wiadomo jakie są)
- $2 + 3 * 5$ jest wyrażeniem arytmetycznym
- można to udowodnić na dwa sposoby
  - albo pierwszym wyrażeniem jest liczba 2 a drugim iloczyn, czyli jakby $2 + (3 * 5)$
  - albo pierwszym wyrażeniem jest suma a drugim liczba, czyli $(2 + 3) * 5$








## Gramatyka, przykład, c.d.

- Wyrażenie $\to$ Liczba | Wyrażenie Operator Wyrażenie
  Operator $\to + \mid - \mid * \mid \div$
  Liczba $\to$ 2 | 3 | 5 (to tylko przykłady liczb)
- możemy wyprowadzić przykładowe wyrażenie
  Wyrażenie $\to$ Wyrażenie Operator Wyrażenie $\Rightarrow$
  Liczba Operator Wyrażenie $\Rightarrow$
  2 Operator Wyrażenie $\Rightarrow$
  $2+$ Wyrażenie $\Rightarrow$
  $2+$ Wyrażenie Operator Wyrażenie $\Rightarrow$
  $2+$ Liczba Operator Wyrażenie $\Rightarrow$
  $2+$ 3 Operator Wyrażenie $\Rightarrow$
  $2+$ $3*$ Wyrażenie $\Rightarrow$
  $2+$ $3*$ Liczba $\Rightarrow$
  $2+$ $3*$ 5





## Gramatyka, definicja

- Gramatyka (bezkontekstowa): $\langle V, T, P, S\rangle$, gdzie
  - $S \in V$ jest wyróżnionym symbolem początkowym
  - $V$ jest zbiorem (skończonym) symboli nieterminalnych (pomocnicze, zmienne, kategorie syntaktyczne)
  - $T$ jest alfabetem, t.j. skończonym zbiorem symboli terminalnych (końcowe, stałe), $V \cap T = \emptyset$
  - $P \subseteq V \times (V \cup T)^*$ jest skończonym zbiorem produkcji zapisywanych jako $v \to w$ gdy $\langle v, w\rangle \in P$
    gdy $\langle v, w_1\rangle \in P, \ldots \langle v, w_n\rangle \in P$ piszemy $v \to w_1 \mid \ldots \mid w_n$
- np. $V = \{W, O, L\}, S = W, T = \{+, -, *, \div, 2, 3, 5\}$ i produkcje
  $W \to L \mid WOW$, $L \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$,
  $O \to + \mid - \mid * \mid \div$
- elementów $\langle V, T, S\rangle$ można nie wypisywać, wiadomo, że symbole nieterminalne są po lewej stronie produkcji, pozostałe są terminalne, pierwszy nieterminalny jest początkowy.





## Gramatyka, wywód słowa

- jeśli $A \to v$ jest produkcją w gramatyce, to $uAw \Rightarrow uvw$ jest krokiem w wywodzie słowa $uvw$ ze słowa $uAw$
- wywody zaczynają się od symbolu początkowego, $S \stackrel{*}{\Rightarrow} w$,
  $w \in (V \cup T)^*$ oznacza, że $S \to w_1 \Rightarrow \ldots \Rightarrow w_n = w$
  - uwaga: zero kroków też jest dopuszczalne $S \stackrel{*}{\Rightarrow} S$
- docelowo chcemy wywieść słowa złożone z symboli terminalnych
- def.: język *definiowany przez gramatykę* $L(G) = \{w \in T^* \mid S \stackrel{*}{\Rightarrow} w\}$
- def.: *język bezkontekstowy* to język generowany przez jakąś gramatykę
- w powyższym przykładzie Wyrażenie $\stackrel{*}{\Rightarrow} 2 + 3 * 5$
- inny przykład $S \to ab \mid aSb$
- język generowany $\{a^n b^n \mid n > 0\}$, nie jest regularny
- gramatyka $S \to aS \mid aB$, $B \to b \mid bB$
- generuje język regularny $\{a^k b^\ell \mid k, \ell > 0\}$

## Gramatyka, wywód słowa, bardziej złożony przykład

- gramatyka: $S \to \varepsilon \mid aB \mid bA$, $A \to aS \mid bAA$, $B \to bS \mid aBB$
- tw.: język generowany to $\{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ – tyle samo $a$ co $b$ w każdym słowie
- dowód przez indukcję względem długości słowa ale dla trzech stwierdzeń jednocześnie
  - $S \stackrel{*}{\Rightarrow} w$ w.t.w. $|w|_a = |w|_b$
  - $A \stackrel{*}{\Rightarrow} w$ w.t.w. $|w|_a = |w|_b + 1$
  - $B \stackrel{*}{\Rightarrow} w$ w.t.w. $|w|_a + 1 = |w|_b$

  analizując każdą z produkcji gramatyki widzimy, że dowód indukcyjny działa
- pamiętamy, że jest to kolejny przykład języka nieregularnego





## Gramatyka, wyrażenie regularne

- wyrażenie regularne (nad $A$) definiowane są gramatyką
  $W \to \emptyset \mid \varepsilon \mid I \mid WW \mid W + W \mid W^* \mid (W)$
  $I \to a$, dla każdej litery $a \in A$
- to nie jest dobra gramatyka, wyrażenia w rodzaju $a + bc^*$ mają wiele wywodów
  gwiazdka może być zastosowana tylko do $c$, do konkatenacji $bc$ albo do całego poprzedzającego wyrażenia
  poprzedzające wyrażenie może być sumą $a$ i $bc$ albo konkatenacją $a + b$ i $c$
- potrzebna będzie lepsza gramatyka





## Operacje na językach bezkontekstowych 1

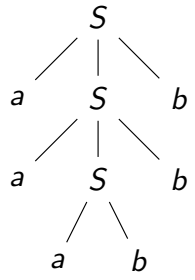
- tw.: klasa języków bezkontekstowych jest zamknięta na sumę, konkatenację i iterację
- konkatenacja i suma: zakładamy, że symbole pomocnicze języków $L_1$ i $L_2$ są różne (np. dodając indeks), dodajemy nowy symbol początkowy $S'$ i nowe produkcje $S' \to S_1 \mid S_2$ dla sumy lub $S' \to S_1 S_2$ dla konkatenacji
  wówczas $S'$ generuje sumę, odpowiednio konkatenację, języków $L_1$ i $L_2$
- iteracja: symbol $S$ generuje język $L$, dodajemy nowy symbol początkowy $S'$ i nową produkcję $S' \to \varepsilon \mid SS'$, $S'$ generuje język $L^*$
- wniosek: każdy język regularny jest bezkontekstowy
- dw.: język regularny definiowany jest wyrażeniem regularnym, zbudowane jest ono używając operacji sumy, konkatenacji i iteracji zaczynając od $\varepsilon$, pojedynczych liter lub $\emptyset$
  wszystkie te konstrukcje są dostępne dla gramatyk, pusty język może być zdefiniowany gramatyką $S \to S$





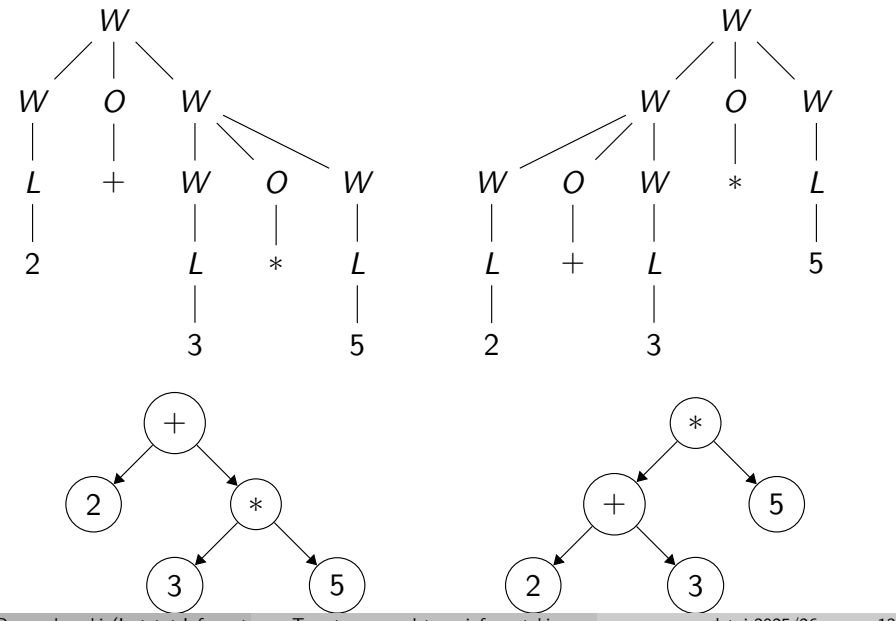
## Drzewo wywodu

- $S \to ab \mid aSb$
  drzewo wywodu słowa $aaabbb$
  $S \to aSb \Rightarrow aaSbb \Rightarrow aaabbb$




- na ogół mamy możliwość rozwijania w danym kroku różnych symboli terminalnych
- zawsze można stosować wywód do symbolu nieterminalnego najbardziej z lewej, albo najbardziej z prawej
- gramatyka $W \to L \mid WOW$, $L \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$, $O \to + \mid - \mid * \mid \div$
- mamy dwa wywody lewostronne słowa $2 + 3 * 5$, było już $W \to WOW \Rightarrow LOW \stackrel{*}{\Rightarrow} 2 + W \Rightarrow 2 + WOW \Rightarrow 2 + LOW \stackrel{*}{\Rightarrow} 2 + 3 * L \Rightarrow 2 + 3 * 5$
- oraz $W \to WOW \Rightarrow WOWOW \Rightarrow LOWOW \stackrel{*}{\Rightarrow} 2 + LOW \stackrel{*}{\Rightarrow} 2 + 3 * W \Rightarrow 2 + 3 * L \Rightarrow 2 + 3 * 5$

## Drzewo wywodu, przykład








## Drzewa wywodu a wywody

- tw.: $S \stackrel{*}{\Rightarrow} w$, $w \in T^*$, wtedy i tylko wtedy gdy istnieje drzewo wywodu z korzeniem $S$ i liśćmi tworzącymi słowo $w$
- dw $\Rightarrow$: jeśli cały wywód kończy się jednym krokiem $S \to w$, to określa on drzewo wysokości 1.
  Jeśli kroków jest więcej, to zaczyna się on $S \to X_1 \dots X_\ell$ i dalej każde z $X_i$ albo jest literą w $w$ albo daje wywód $X_i \stackrel{*}{\Rightarrow} w_i$ do fragmentu słowa $w$. Każdy z tych wywodów daje się opisać drzewem z korzeniem $X_i$ i liśćmi $w_i$, wszystkie te drzewa składają na szukane drzewo.
- dw $\Leftarrow$: Jeśli drzewo ma wysokość 1, to znaczy, że wszystkie litery w $w$ są wyprowadzane w jednej produkcji $S \to w$.
  Jeśli drzewo jest wyższe, to korzeń określa wywód $S \to X_1 \dots X_\ell$, dla liści z poddrzewa stosujemy założenie indukcyjne i znajdujemy wywody $X_i \stackrel{*}{\Rightarrow} w_i$. Stosując je po kolei od lewego, albo i w dowolnej kolejności, otrzymujemy wywód $S \stackrel{*}{\Rightarrow} w$.





## Gramatyki jednoznaczne

- def. gramatyka jest *jednoznaczna*, jeśli każde słowo generowane przez tę gramatykę ma tylko jeden wywód lewostronny
  - równoważnie: każde słowo ma jednoznaczne drzewo wywodu
- gramatyka poprzednia nie jest jednoznaczna
- można jednak podać równoważną gramatykę jednoznaczną
- np. wymusić by działania były wykonywane od razu od lewej (tak jak na prostym kalkulatorze)
- albo by mnożenie miało wyższą siłę wiązania (tak jak na ogół przyjmujemy dla wyrażeń arytmetycznych i podobnych)



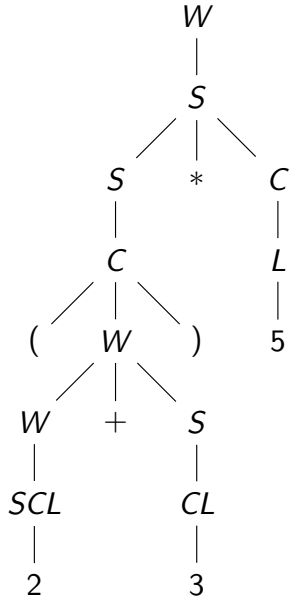


## Gramatyki jednoznaczne, przykład $(2 + 3) * 5$

- gramatyka jednoznaczna dla prostych wyrażeń arytmetycznych (tylko dodawanie i mnożenie, mnożenie ma większą siłę wiązania, wiązanie jest do lewej):

| | |
|---|---|
| $W \to S \mid W + S$ | wyrażenie |
| $S \to C \mid S * C$ | składnik |
| $C \to L \mid (W)$ | czynnik |
| $L \to 0 \mid \dots \mid 9$ | liczba |

- napis $2 + 3 * 5$ ma jednoznaczne drzewo wywodu zaczynające się od $W \to W + S$, mnożenie $3 * 5$ nie musi być brane w nawias
- druga interpretacja wymaga nawiasów, wywód $(2 + 3) * 5$ zaczyna się od $W \to S \to S * C$
- ta gramatyka dopuszcza niepotrzebne nawiasy



## Gramatyki jednoznaczne, wyrażenia regularne

- gramatyka jednoznaczna dla wyrażeń regularnych: siła wiązania nawiasy $>$ iteracja $>$ konkatenacja $>$ suma
  wiązanie jest do lewej, np. $abc$ oznacza de facto $(ab)c$, co semantycznie nie ma znaczenia, bo konkatenacja jest łączna

| | |
|---|---|
| $W \to S \mid W + S$ | wyrażenie |
| $S \to C \mid SC$ | składnik |
| $C \to I \mid I^*$ | czynnik |
| $I \to L \mid (W)$ | iteracja |
| $L \to a$ | każda litera $a \in A$ |

- wyrażenie $a + bc^*$ ma wywód którego fragmenty to $I \to L \to c$, $C \overset{*}{\Rightarrow} c^*$, $S \overset{*}{\Rightarrow} bc^*$ i na końcu $W \overset{*}{\Rightarrow} a + bc^*$
- inne, wspomniane przedtem, próby wywodu tego wyrażenia w tej gramatyce są niemożliwe





## Gramatyki jednoznaczne, c.d.

- istnieją języki bezkontekstowe nie posiadające jednoznacznej gramatyki, językiem takim jest

$$\{a^n b^n c^\ell \mid n, \ell \geq 1\} \cup \{a^n b^\ell c^\ell \mid n, \ell \geq 1\}$$

  dowód jest poza zasięgiem tego wykładu
- słowo $a^n b^n c^n$ pasuje do obu wzorców i ma niejednoznaczny wywód w oczywistej gramatyce
- wszystkie znane przykłady wymagają by w powyższym przykładzie przekrój występujących języków *nie* był bezkontekstowy

$$\{a^n b^n c^\ell \mid n, \ell \geq 1\} \cap \{a^n b^\ell c^\ell \mid n, \ell \geq 1\} = \{a^n b^n c^n \mid n \geq 1\}$$

  zobaczymy, że ten przekrój nie jest językiem bezkontekstowym





## Upraszczanie gramatyk

- nie chcemy w gramatyce zmiennych, które nie pojawią się nigdy w wywodach słów generowanego języka (o ile jest on niepusty)
- tw.: dla każdej gramatyki $G$ istnieje $G''$ taka, że $L(G) = L(G'')$ i dla każdego $A \in V''$, $A \overset{*}{\Rightarrow} w$ dla pewnego $w \in T^*$ oraz $S \overset{*}{\Rightarrow} uAv$ dla pewnych $u, v \in (T \cup V'')^*$
- dw.: budujemy $V'$ iteracyjnie dodając te symbole $A$, dla których $A \to X_1 \dots X_\ell$ i wszystkie $X_i$ są albo terminalne albo już w $V'$
  w kolejnym kroku kroku budujemy $V''$ zaczynając od $S$ i umieszczając tam wyłącznie zmienne występujące w produkcjach angażujących zmienne z $V'$
- przykład: $S \to AB \mid c$, $A \to a$
  $V' = \{S, A\}$, pozostaje $S \to c, A \to a$
  $V'' = \{S\}$, pozostaje $S \to c$, $T'' = \{c\}$, $L = \{c\}$





## Upraszczanie gramatyk, c.d.

- tw.: $L \setminus \{\varepsilon\}$ jest generowany przez gramatykę, w której nie występuje $\varepsilon$
- dw.: $A \in V$ jest zerowalny, jeśli $A \overset{*}{\Rightarrow} \varepsilon$
  - dla każdej produkcji $A \to X_1 \dots X_\ell$, jeśli jakiś symbol $X_i$ jest zerowalny, dołączamy produkcję tę samą ale z $X_i$ usuniętym (czyli zastąpionym przez $\varepsilon$), rozpatrujemy wszystkie podzbiory zerowalnych zmiennych z wyjątkiem wszystkich naraz
  - usuwamy produkcje $A \to \varepsilon$
  - $A \overset{*}{\Rightarrow} w$ w nowej gramatyce w.t.w. $A \overset{*}{\Rightarrow} w$ w starej i $w \neq \varepsilon$
- czyli jeśli $\varepsilon \in L$, to wystarczy by w gramatyce była produkcja $S \to \varepsilon$, pozostałe produkcje nie muszą wspominać o $\varepsilon$
  całkowity brak produkcji $A \to \varepsilon$ oznacza, że w każdym wywodzie słowa mają długości niemalejące, jednak $S \to \varepsilon$ oraz wystąpienie $S$ w dalszych produkcjach sprawi jeszcze problem

## Upraszczanie gramatyk, III

- niczego nie produkują produkcje $A \to B$ (jednostkowe), $B \in V$, nie dotyczy to możliwych produkcji $A \to a$, $a \in T$
- tw.: każdy $L \setminus \{\varepsilon\}$ jest definiowany przez gramatykę bez symboli bezużytecznych, bez produkcji $\to \varepsilon$ oraz bez produkcji jednostkowych
- dw.: zakładamy najpierw, że $\varepsilon \notin L$, na początku zostawiamy wszystkie produkcje niejednostkowe
  jeśli $A \stackrel{*}{\Rightarrow} B \to \alpha$, to dołączamy produkcję $A \to \alpha$ (wszystkie kroki w wywodzie $A \stackrel{*}{\Rightarrow} B$ są jednostkowe)
  następnie stosujemy wcześniejsze algorytmy do oczyszczenia gramatyki, być może dodając $S \to \varepsilon$

gramatyka po "uproszczeniu" da prostsze drzewa wywodu, ale niekoniecznie jest bardziej czytelna

| | |
|---|---|
| $W \to 0 \mid \ldots \mid 9 \mid (W) \mid S * C \mid W + S$ | wyrażenie |
| $S \to 0 \mid \ldots \mid 9 \mid (W) \mid S * C$ | składnik |
| $C \to 0 \mid \ldots \mid 9 \mid (W)$ | czynnik |





## Postać normalna Chomsky'ego

- gramatyka $\langle V, T, P, S\rangle$ jest w postaci normalnej Chomksy'ego jeśli wszystkie produkcje są jednej z postaci
  - $A \to a$, gdzie $A \in V$, $a \in T$
  - $A \to BC$, gdzie $B, C \in V \setminus S$
  - $S \to \varepsilon$

  w szczególności nie ma produkcji jednostkowych, nie ma $\varepsilon$ produkcji poza być może początkową, symbol końcowy nie występuje po prawej stronie
- tw.: dla każdej gramatyki bezkontekstowej istnieje równoważna w postaci normalnej Chomsky'ego
  zawsze można też zażądać, by nie było symboli bezużytecznych





## Postać normalna Chomsky'ego, dowód

- możemy założyć $\varepsilon \notin L$, na końcu będzie czas dodać $S \to \varepsilon$
  możemy założyć, że gramatyka nie zawiera produkcji jednostkowych ani produkcji $\varepsilon$
  każda produkcja $A \to a$ jest już we właściwym formacie
- dla każdej litery $a \in T$ definiujemy nowy symbol pomocniczy $C_a \in V'$ i produkcję $C_a \to a$
  rozpatrzmy produkcję $A \to X_1 \ldots X_\ell$, być może niektóre $X_i \in T$, zastępujemy je ich odpowiednikami w $V'$, każdy wywód $A \to w$ w starej gramatyce można zastąpić wywodem w nowej i vice versa
- zamiast produkcji $A \to X_1 \ldots X_\ell$, $X_i \in V'$, $\ell > 2$, wprowadzamy nowe zmienne $B_2, \ldots, B_{\ell-1}$ i nowe produkcje
  $A \to X_1 B_2$, $B_2 \to X_2 B_3, \ldots, B_{\ell-1} \to X_{\ell-1} X_\ell$
- jeśli symbol początkowy występuje w którejkolwiek produkcji, to wprowadzamy nowy symbol, produkcję $S' \to S$ i wykonujemy algorytm eliminacji produkcji jednostkowych





## Postać normalna Greibach

- gramatyka $\langle V, T, P, S\rangle$ jest w postaci normalnej Greibach jeśli wszystkie produkcje są jednej z postaci
  - $A \to aA_1 \ldots A_n$, gdzie $a \in T$, $A_1, \ldots, A_n \in V \setminus S$, $n \geq 0$
  - $S \to \varepsilon$

  w szczególności symbol końcowy nie występuje po prawej stronie
- tw.: dla każdej gramatyki bezkontekstowej istnieje równoważna w postaci normalnej Greibach

## Lemat o pompowaniu

- Tw.: jeśli $L$ jest językiem bezkontekstowym, to istnieje stała $n$ taka, że dla każde słowo $z \in L$ dłuższe niż $n$ posiada rozkład $z = uvwxy$ taki, że
  - $|vwx| \leq n$
  - $|v| + |x| \geq 1$
  - $uv^\ell wx^\ell y \in L$ dla wszystkich $\ell \in \mathbb{N}$
- przykład zastosowania: język $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ nie jest językiem bezkontekstowym
  albo fragment pompowany jest na granicy liter i po napompowaniu będzie zaburzona kolejność,
  albo pompowane będą jednorodne fragmenty, ale najwyżej dwa, a w słowach są trzy





## Lemat o pompowaniu, dowód

- gramatyka posiada skończoną liczbę symboli nieterminalnych $k$, znany jest stopień rozgałęzienia drzew wywodu (np. można przyjąć postać normalną Chomsky'ego) czyli wywód słowa $z$ długości $\geq 2^k$ wymaga powtórzenia się jakiegoś symbolu na jednej ze ścieżek
- rozważmy ostatnie i przedostatnie wystąpienie symbolu $X$, w ostatnim wystąpieniu $X \stackrel{*}{\Rightarrow} w$, w przedostatnim wystąpieniu $X \stackrel{*}{\Rightarrow} vwx$
- na pewno $|v| + |x| \geq 1$, jeśli $n$ jest wybrane jako minimalna liczba gwarantująca powtórzenie, to $|vwx| \leq n$
- słowo wyprowadzone w ostatnim wystąpieniu można podstawić w przedostatnie, czyli $uwy \in L$, i na odwrót, przedostatnie wystąpienie można wkleić w miejsce ostatniego wystąpienia otrzymując $uv^2wx^2y \in L$, to podstawienie można iterować dowolną liczbę razy





## Lemat o pompowaniu, przykład

- wyrażenie $(2 + 3) * 5$ ma wywód
  $W \stackrel{*}{\Rightarrow} S * 5 \stackrel{*}{\Rightarrow} (W) * 5 \stackrel{*}{\Rightarrow} (2 + S) * 5 \stackrel{*}{\Rightarrow} (2 + 3) * 5$
  ostatnie wystąpienie $S$ rozwija się do liczby 3, przedostatnie rozwija się do $(2 + 3)$
  po zastąpieniu ostatniego wystąpienie przedostatnim otrzymujemy $(2 + (2{+}3)) * 5$
- czyli $u = \varepsilon, v = (2 +, w = 3, x =), y = \ * 5$