

Stefan Sokołowski

JĘZYKI FORMALNE I METODY KOMPILACJI

wykład 1

Inst. Informatyki Stosowanej, ANS
Elbląg, 2025/2026

Wykład 1, str. 1

Języki formalne

JĘZYK ŹRÓDŁOWY
(wysokiego poziomu)



JĘZYK DOCELOWY
(wewnętrzny)

Kompilator musi „znać” oba języki...

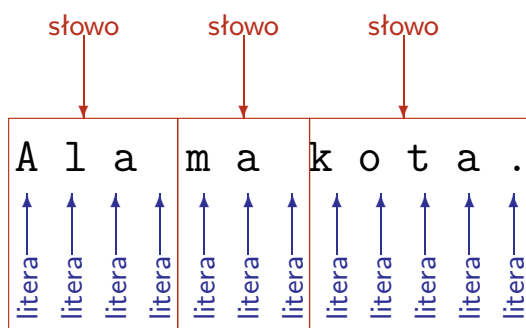
Jak program komputerowy nauczyć języka?
Najpierw trzeba ten język ściśle **zdefiniować**.

Języki formalne mają u podstaw sposoby ścisłego definiowania języków.

Języki formalne

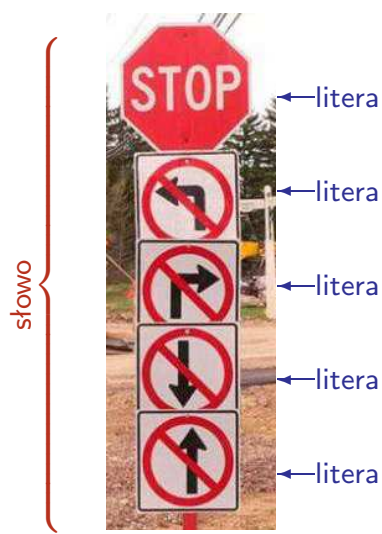


PRZYKŁAD:



Języki formalne

PRZYKŁADY:



litery:

```
class Test {
    public static void main
        ( String [ ] args ) {
        System . out . println
            ( "To tylko test." ) ;
    }
}
```

Proste języki formalne (przykłady)

$\{a, \text{aback}, \text{abandon}, \text{abase}, \dots, \text{zoom}\}$

— język (**skończony**) składający się ze słów języka angielskiego

$\{0, 1, 10, 11, 100, 101, \dots\}$

— język (**nieskończony**) składający się z liczb binarnych

$\{a^n b^n \mid n \geq 0\} = \{\lambda, ab, aabb, aaabbb, \dots\}$

puste (**to nie jest litera!**)

— język (**nieskończony**) składający się z takich słów, w których

- występują tylko litery a i b ,
- liczba liter a jest równa liczbie liter b ,
- wszystkie litery a poprzedzają wszystkie litery b .

Języki formalne

DEFINICJE:

Σ — skończony niepusty *alfabet*, czyli zbiór tzw. *liter*

słowo — dowolny ciąg skończonej długości o elementach z Σ

Σ^* — zbiór **wszystkich** słów

język — dowolny zbiór słów $L \subseteq \Sigma^*$

PRZYKŁAD: nie mylić! —

λ — słowo *puste* (ciąg liter o długości 0)

Uwaga: to nie jest litera!

$\emptyset \stackrel{\text{def}}{=} \{\}$ — język pusty (nie zawierający żadnego słowa)

$\Lambda \stackrel{\text{def}}{=} \{\lambda\}$ — język zawierający wyłącznie słowo puste

PRZYKŁAD:

$1(*2-1)(1-(2-*1))()$ — słowo nad alfabetem $\Sigma \stackrel{\text{def}}{=} \{ (,), -, *, 1, 2 \}$

$\{\lambda, a, b, ab, ba\}$ — język nad alfabetem $\Sigma \stackrel{\text{def}}{=} \{a, b\}$ zawierający pięć słów

$\{a^n \mid n \geq 0\} = \{\lambda, a, aa, aaa, \dots\}$ — jęz. nieskończ. nad alf. $\Sigma \stackrel{\text{def}}{=} \{a\}$

Jak **porządnie** opisać język nieskończony?

- $\{\lambda, ab, aabb, aaabbb, \dots\}$ — co oznaczają kropeczki?
- $\{a^n b^n \mid n \geq 0\}$ — lepiej, ale są niepożądane elementy obce (liczby, zmienne, relacje, ...)

Język definiujemy na dwa sposoby. Przez

- **generator** — cokolwiek **wyprodukuje**, uważamy za należące do języka
- **akceptor** — cokolwiek **zatwierdzi**, uważamy za należące do języka

Opis języka programowania przeznaczony dla programisty powinien być **generatorem** — ma wyjaśniać, jak pisać poprawne programy.

Kompilator języka programowania powinien zawierać **akceptor** — sprawdzający, czy program na jego wejściu należy do języka.

Generatory: gramatyki bezkontekstowe

DEFINICJA: Gramatyka bezkontekstowa G — czwórka $\langle \Sigma, N, P, S \rangle$

- alfabet *terminalny* Σ ;
- alfabet *nieterminalny* lub *pomocniczy* N (rozłączny z Σ);
- skończony zbiór P *produkcji* postaci $A \rightarrow w$, gdzie
 $A \in N$ (czyli jest pojedynczym nieterminaliem),
 $w \in (N \cup \Sigma)^*$ (czyli jest słowem z terminali i nieterminali);
- wybrany nieterminal $S \in N$ nazywany *nieterminaliem początkowym* albo *aksjomatem* gramatyki.

PRZYKŁAD:

Terminale: a, b

Nieterminale: X

Aksjomat: X

Produkcje:

$$\begin{array}{l} X \rightarrow \lambda \\ X \rightarrow aXa \\ X \rightarrow bXb \end{array}$$

Jak gramatyka definiuje język?

DEFINICJA: Wywodliwość —

$uXv \Rightarrow_G u w v$ czyli $u w v$ jest *bezpośrednio wywodliwe* z uXv $\iff$ w P jest produkcja $X \rightarrow w$

$w_1 \Rightarrow_G^* w_n$ czyli w_n jest *wywodliwe* z w_1 $\iff$ $w_1 \Rightarrow_G w_2 \Rightarrow_G w_3 \Rightarrow_G \dots \Rightarrow_G w_{n-1} \Rightarrow_G w_n$

DEFINICJA: Język $L(G)$ *generowany przez gramatykę* G — składa się z tych słów wywodliwych z **aksjomatu**, które już nie zawierają nieterminali: $L(G) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$.

PRZYKŁAD:

Gramatyka:

$X \rightarrow \lambda$
 $X \rightarrow aXa$
 $X \rightarrow bXb$

$X \Rightarrow_G^* bbaabb \in L(G)$

LEMAT:

$L(G) = \{ww^R \mid w \in \{a,b\}^*\}$

$X \Rightarrow bXb \Rightarrow bbXbb \Rightarrow bbaXabb \Rightarrow bbaabb$

Jak gramatyka definiuje język?

NIE MYLIĆ:

$\rightarrow$ — formalny symbol w produkcji w gramatyce

$\Rightarrow$ — relacja bezpośredniej wywodliwości:

jeśli $X \rightarrow w$ to $X \Rightarrow w$

ale np. $0X1 \Rightarrow 00X11$

a **nieprawda**, że $0X1 \rightarrow 00X11$

$\Rightarrow^*$ — relacja wywodliwości:

jeśli $w_1 \Rightarrow w_2$ to $w_1 \Rightarrow^* w_2$

ale np. $X \Rightarrow^* 0011$

a **nieprawda**, że $X \Rightarrow 0011$

$X \rightarrow \lambda$
 $X \rightarrow 0X1$

Jak gramatyka definiuje język?

PRZYKŁAD:

Gramatyka G :

$$\begin{array}{lcl} E & \rightarrow & T \\ E & \rightarrow & E + T \\ T & \rightarrow & F \\ T & \rightarrow & T * F \\ F & \rightarrow & a \\ F & \rightarrow & (E) \end{array}$$

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \\ &\Rightarrow a + T * F \Rightarrow a + F * F \Rightarrow a + a * F \\ &\Rightarrow a + a * a \in L(G) \end{aligned}$$

Jak gramatyka definiuje język?

PRZYKŁAD:

Gramatyka G :

$$\begin{array}{lcl} E & \rightarrow & T \\ E & \rightarrow & E + T \\ T & \rightarrow & F \\ T & \rightarrow & T * F \\ F & \rightarrow & a \\ F & \rightarrow & (E) \end{array}$$

$$\begin{aligned} E &\Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (E + T) \Rightarrow (T + T) \\ &\Rightarrow (F + T) \Rightarrow (a + T) \Rightarrow (a + F) \\ &\Rightarrow (a + a) \in L(G) \end{aligned}$$

$$L(G) = \text{wyrażenia arytmetyczne zbudowane z } a, +, *, (,)$$