

Formális nyelvek a gyakorlatban

Formális nyelvek, 1. gyakorlat

Célja: A programozási nyelvek szintaxisának leírására használatos eszközök, módszerek bemutatása

Fogalmak: BNF, szabály, levezetés, leírt nyelv, EBNF, szintaxis-diagram, EP programozási nyelv, Pascal programozási nyelv

Feladatok jellege: Egyszerűbb programnyelvi konstrukciók BNF leírása, majd átírása EBNF-re, az EP megadása EBNF-el (előre kiadott lapon), program készítés EP-ben, a PASCAL szintaxis-gráfokon alapuló leírásának áttekintése (előre kiadott lapokon), bizonyos konstrukciók átírása EBNF-re.

2005/06 II. félév

BNF

BNF definíciója

BNF: **Backus-Naur Form**

Metanyelv, melynek segítségével szabályok alkothatók meg.

Építőkövei:

- $\langle \text{név} \rangle$ (**fogalmak**)
- $::=$ a szabály bal- és jobboldalának elválasztására szolgál.
- a definiálandó nyelv karakterkészlete (**terminálisok**)

Szabályok

Egy **szabály** bal- és jobboldalból áll, köztük $::=$, baloldalon pontosan 1 fogalom, jobboldalon terminálisok és fogalmak sorozata lehet.

Példa BNF-re: $\langle \text{azonosító} \rangle$

Szabályok

$\langle \text{azonosító} \rangle ::= \langle \text{betű} \rangle$
 $\langle \text{azonosító} \rangle ::= \langle \text{betű} \rangle \langle \text{azonosítótég} \rangle$
 $\langle \text{azonosítótég} \rangle ::= \langle \text{betű} \rangle$
 $\langle \text{azonosítótég} \rangle ::= \langle \text{számjegy} \rangle$
 $\langle \text{azonosítótég} \rangle ::= \langle \text{betű} \rangle \langle \text{azonosítótég} \rangle$
 $\langle \text{azonosítótég} \rangle ::= \langle \text{számjegy} \rangle \langle \text{azonosítótég} \rangle$
 $\langle \text{betű} \rangle ::= a$
:
 $\langle \text{betű} \rangle ::= z$
 $\langle \text{számjegy} \rangle ::= 0$
:
 $\langle \text{számjegy} \rangle ::= 9$

Példa BNF-re: $\langle \text{azonosító} \rangle$

Levezetés

Milyen értelemben ír le azonosítót?

Pl. **a12** azonosító-e?

$\langle \text{azonosító} \rangle \rightarrow \langle \text{betű} \rangle \langle \text{azonosítótég} \rangle \rightarrow$
 $a \langle \text{azonosítótég} \rangle \rightarrow a \langle \text{számjegy} \rangle \langle \text{azonosítótég} \rangle \rightarrow$
 $a1 \langle \text{azonosítótég} \rangle \rightarrow a1 \langle \text{számjegy} \rangle \rightarrow a12$

Ekkor azt mondjuk, hogy az $\langle \text{azonosító} \rangle$ fogalomból **levezethető a12**.

u helyes azonosító \Leftrightarrow **u** előállítható az $\langle \text{azonosító} \rangle$ fogalomból a fenti helyettesítési mechanizmussal.

Példa BNF-re: \langle azonosító \rangle

Csoportosítás

Egyszerűbb írásmód: alternatívák, kiemelések (csoport)

```
 $\langle$ azonosító $\rangle ::= \langle$ betű $\rangle | \langle$ betű $\rangle \langle$ azonosítóvég $\rangle$   
 $\langle$ azonosító $\rangle ::= \langle$ betű $\rangle \{ | \langle$ azonosítóvég $\rangle \}$   
 $\langle$ azonosítóvég $\rangle ::= \langle$ betű $\rangle | \langle$ számjegy $\rangle | \langle$ betű $\rangle \langle$ azonosítóvég $\rangle |$   
                           $\langle$ számjegy $\rangle \langle$ azonosítóvég $\rangle$   
 $\langle$ azonosítóvég $\rangle ::= \{ \langle$ betű $\rangle | \langle$ számjegy $\rangle \} \{ | \langle$ azonosítóvég $\rangle \}$   
 $\langle$ betű $\rangle ::= a | b | \dots | z$   
 $\langle$ számjegy $\rangle ::= 0 | 1 | \dots | 9$ 
```

Az ismétlődéseket a BNF rekurzióval kezeli!

Pl. \langle azonosítóvég \rangle

EBNF

BNF kiterjesztése

EBNF: **Extended Backus-Naur Form**
iteratív változat

```
 $\langle$ azonosító $\rangle ::= \langle$ betű $\rangle @ \{ \langle$ betű $\rangle | \langle$ számjegy $\rangle \}$   
akárhány lépésű iteráció.
```

```
 $\langle$ azonosító $\rangle ::= \langle$ betű $\rangle \{ \langle$ betű $\rangle | \langle$ számjegy $\rangle \}_0^5$   
legfeljebb 5, legalább 0 ismétlődés.
```

EBNF előnyei

Példa: Valós számok BNF-fel

```
 $\langle$ törtszám $\rangle ::= \langle$ egész $\rangle . | \langle$ egész $\rangle . \langle$ egész $\rangle | . \langle$ egész $\rangle$   
 $\langle$ egész $\rangle ::= \langle$ számjegy $\rangle | \langle$ számjegy $\rangle \langle$ egész $\rangle$   
 $\langle$ számjegy $\rangle ::= 0 | 1 | \dots | 9$   
 $\langle$ előjel $\rangle ::= | + | -$   
 $\langle$ valósszám $\rangle ::= \langle$ egész $\rangle E \langle$ számjegy $\rangle |$   
                   $\langle$ egész $\rangle E \langle$ előjel $\rangle \langle$ számjegy $\rangle \langle$ számjegy $\rangle |$   
                   $\langle$ törtszám $\rangle |$   
                   $\langle$ törtszám $\rangle E \langle$ előjel $\rangle \langle$ számjegy $\rangle |$   
                   $\langle$ törtszám $\rangle E \langle$ előjel $\rangle \langle$ számjegy $\rangle \langle$ számjegy $\rangle$ 
```

EBNF előnyei

Példa: Valós számok **EBNF**-fel

```
 $\langle$ törtszám $\rangle ::= \langle$ egész $\rangle . \{ | \langle$ egész $\rangle \} | . \langle$ egész $\rangle$   
 $\langle$ egész $\rangle ::= \langle$ számjegy $\rangle @ \langle$ számjegy $\rangle$   
 $\langle$ számjegy $\rangle ::= 0 | 1 | \dots | 9$   
 $\langle$ valósszám $\rangle ::= \langle$ törtszám $\rangle |$   
 $\{ \langle$ törtszám $\rangle | \langle$ egész $\rangle \} E \{ | + | - \} \langle$ számjegy $\rangle \{ | \langle$ számjegy $\rangle \}$ 
```

Az EP nyelv leírása EBNF-fel/1

```
<program> ::= <utasítás>;@ { <utasítás>; }
<utasítás> ::= { <címke>: | } <törzs>
<törzs> ::= { READ | WRITE } <azonosító> |
          LET <azonosító> = <kifejezés> |
          GOTO <címke> { IF <kifejezés> | }
<azonosító> ::= <betű> { <számjegy> }01
<betű> ::= X | Y | Z
<számjegy> ::= 0 | 1 | ... | 9
```

Az EP nyelv leírása EBNF-fel/2

```
<kifejezés> ::= { + | - | } <előjeltelen kifejezés>
<előjeltelen kifejezés> ::= <tag> @ { { + | - } <tag> }
<tag> ::= <tényező> @ { { * | / } <tényező> }
<tényező> ::= <azonosító> | <szám> | ( <kifejezés> )
<címke> ::= <szám>
<szám> ::= <számjegy> { <számjegy> }01
```

Fehér szóközök, kocsi vissza/soremelés karakterek.
IF: ha a kifejezés ≥ 0 , ugrik.

Példa EP-vel megoldható feladatra

Legnagyobb közös osztó

Írjunk olyan EP programot, mely beolvasson két pozitív egész számot, majd kiírja legnagyobb közös osztóját!

Q: $x = x' > 0$ $y = y' > 0$ egészek

R: $x = \text{luko}(x', y')$

Ezt megvalósító algoritmus:

$x \neq y$	
$x > y$	$x < y$
$x := x - y$	$y := y - x$

Példa EP-vel megoldható feladatra

EP program luko-ra

```
READ X;
READ Y;
100: GOTO 101 IF X-Y;
      LET Y=Y-X;
      GOTO 100;
101: GOTO 102 IF Y-X;
      LET X=X-Y;
      GOTO 100;
102: WRITE X;
```

EBNF használata adatszerkezetek leírására

Bináris fák, listák

Bináris fák:

\overline{NIL} üres objektum
Bináris fa $\left\{ \begin{array}{l} \overline{NIL} \\ [\text{gyökér, baloldal, jobboldal}] \end{array} \right.$

$\langle \text{egész bin. fa} \rangle ::= \overline{NIL} \mid [\langle \text{egész} \rangle, \langle \text{egész bin. fa} \rangle, \langle \text{egész bin. fa} \rangle]$

$\langle \text{valós bin. fa} \rangle ::= \overline{NIL} \mid [\langle \text{valós} \rangle, \langle \text{valós bin. fa} \rangle, \langle \text{valós bin. fa} \rangle]$

$\langle \text{egész lista} \rangle ::= \overline{NIL} \mid [\langle \text{egész} \rangle, \langle \text{egész lista} \rangle]$

$\langle \text{valós lista} \rangle ::= \overline{NIL} \mid [\langle \text{valós} \rangle, \langle \text{valós lista} \rangle]$

EBNF használata adatszerkezetek leírására

W nyelvtan

Módosítás:

$\langle \hat{X} \text{ lista} \rangle ::= \overline{NIL} \mid [\langle \hat{X} \rangle, \langle \hat{X} \text{ lista} \rangle]$

$\hat{X} := \text{egész} \mid \text{valós} \mid \text{Boole} \mid \text{komplex}$

Hiperszabály

Kétszintű EBNF

van Wijngaarden nyelvtan (W nyelvtan)

$\langle \hat{X} \text{ bin. fa} \rangle ::= \overline{NIL} \mid [\langle \hat{X} \rangle, \langle \hat{X} \text{ bin. fa} \rangle, \langle \hat{X} \text{ bin. fa} \rangle]$

Szintaxis gráfok

Definíció

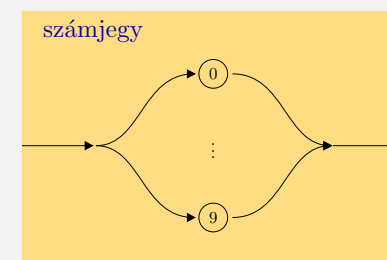
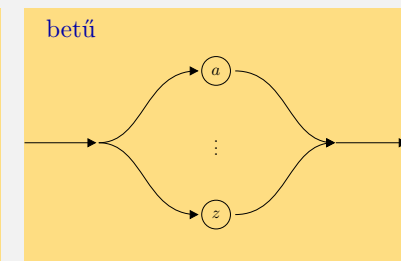
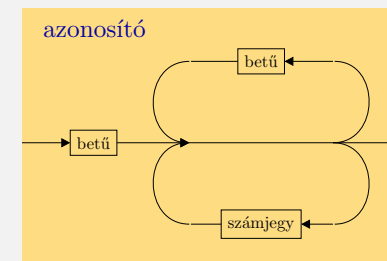
Az EBNF-fel történő leírás szemléletesebb, gráfok segítségével történő megadása.

- A fogalmak a \square jelű csomópontok, bennük a fogalom neve,
- a terminálisok a \circ jelű csomópontok, bennük egy terminális sorozat,
- a gráf neve a definiálandó fogalom.

A gráf irányított, összefüggő, egy bemeneti és egy kimeneti éllel.

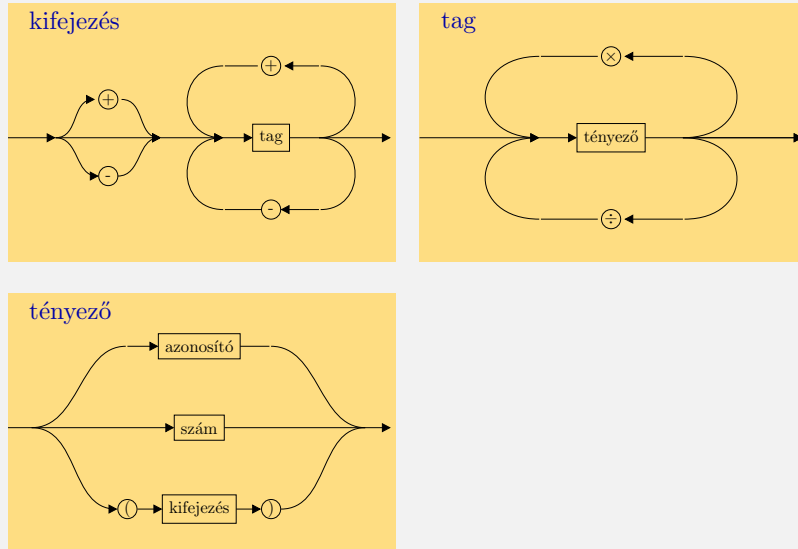
Szintaxis gráfok

Példa: Azonosító



Szintaxis gráfok

Példa: EP nyelv



Házi feladat

1. Módosított azonosító: belsejében lehet _ jel is. Kezdődhet, de nem végződhet vele, két aláhúzás nem lehet egymás mellett. Írjuk fel (E)BNF formulákkal!
2. EP teljes átírása szintaxis gráfokkal.
3. A kiadott PASCAL szintaxis tanulmányozása. A kifejezések átírása (E)BNF-re.
4. Kifejezések leírása W nyelvtannal.