

Formális nyelvek a gyakorlatban

Formális nyelvek, 1. gyakorlat

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

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

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

2008/09 I. félév

Segédanyagok

1. Dr. Hunyadvári László, Manhertz Tamás: *Automaták és formális nyelvek*
<http://aszt.inf.elte.hu/~hunlaci/book.pdf>
2. Fülöp Zoltán: *Formális nyelvek és szintaktikus elemzésük* (Polygon)
3. Bach Iván: *Formális nyelvek* (Typotex)
4. Gyakorlatfóliák (ahogy a gyakorlat előrehalad)
<http://www.cs.elte.hu/~tichlerk>

Fordítóprogramok

A formális nyelvek elméletével az informatikusok többek között a fordítóprogramok működésének megértése miatt foglalkoznak.

Mit csinál egy fordítóprogram?

- ellenőrzi, hogy helyes-e a program
- ha helyes lefordítja

Mit értünk az alatt, hogy helyes?

Teljesít bizonyos szintaktikai feltételeket. (Például adott kulcsszóval kezdődik, adott kulcsszavakat használ adott szabályok szerint, az utasítások között valamilyen elhatároló jel áll, a végén egy adott jel szerepel.)

Szintaktikai helyesség

Ha egy program teljesíti a szintaxisban megadott szabályokat **szintaktikailag helyesnek** nevezzük.

A szintaktikai elemzés alapeladata: Döntsük el egy programról, hogy szintaktikailag helyes-e.

A programozási nyelvek helyett ennél absztraktabb fogalommal, a formális nyelvekkel fogunk foglalkozni. Tárgyaljuk nyelvek megadásának módszereit – melyek közül a legfontosabbak a **generatív nyelvtanok** és az **automaták** lesznek –, osztályozzuk őket és megpróbálunk a fenti problémára is válaszolni.

Történeti megjegyzés: A matematikai nyelvészet alapjait 1956-ban Noam Chomsky rakta le. Az eredeti motiváció a természetes nyelvek gépi fordítása volt.

Példa a szintaxis leírására: BNF

BNF definíciója

BNF: Backus-Naur Form John Backus, Peter Naur ALGOL 60

A BNF egy széles körben használt metanyelv melynek segítségével szabályok alkothatók meg (például egy programozási nyelv szabályai).

Építőkövei:

- $\langle \text{név} \rangle$ (**fogalmak** vagy más néven **nemterminálisok**)
- $::=$ a szabály bal- és jobboldalának elválasztására szolgáló jel.
- a definiálandó nyelv karakterkészlete, a sztringek alkotóelemei (**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

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

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

Levezetés

Milyen értelemben ír le azonosítót?

Pl. a^{12} azonosító-e?

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

Minden lépésben egy fogalmat helyettesítettünk egy olyan szabály jobboldalával, melynek a baloldalán az adott fogalom áll. Ekkor azt mondjuk, hogy az $\langle \text{azonosító} \rangle$ fogalomból **levezethető** a^{12} .

Terminálisok egy u sorozata helyes azonosító $\Leftrightarrow u$ előállítható az $\langle \text{azonosító} \rangle$ fogalomból a fenti helyettesítési mechanizmussal, azaz létezik legalább egy levezetése az $\langle \text{azonosító} \rangle$ fogalomból.

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

Csoportosítás

Egyszerűbb írásmód: alternatívák, kiemelések (csoport)

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

Az ismétlődéseket a BNF rekurzióval kezeli!

Pl. $\langle \text{azonosítóvég} \rangle$

$\langle \text{azonosítóvég} \rangle \rightarrow \langle \text{betű} \rangle \langle \text{azonosítóvég} \rangle \rightarrow \langle \text{betű} \rangle \langle \text{betű} \rangle \langle \text{azonosítóvég} \rangle \rightarrow$
 $\rightarrow \langle \text{betű} \rangle \langle \text{betű} \rangle \langle \text{betű} \rangle \langle \text{azonosítóvég} \rangle \rightarrow \dots$

EBNF

BNF kiterjesztése

EBNF: Extended Backus-Naur Form
iteratív változat

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

A „@” utáni terminálist, nemterminálist vagy csoportot akárhányszor ismételhetjük (a 0-t is beleértve).

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

A terminálist, nemterminálist vagy csoportot legalább illetve legfeljebb annyiszor ismételhetjük mint az alsó illetve felső indexe.

Ami EBNF-fel leírható az BNF-fel is, de EBNF-fel esetleg egyszerűbben, tömörebben.

EBNF előnyei

Példa: Nemnegatív valós számok BNF-fel

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

EBNF használatának előnyei

Példa: Nemnegatív valós számok EBNF-fel

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

Az EP programozási nyelv leírása EBNF-fel/1

```
 $\langle \text{program} \rangle ::= \langle \text{utasítás} \rangle ; @ \{ \langle \text{utasítás} \rangle ; \}$   
 $\langle \text{utasítás} \rangle ::= \{ \langle \text{címke} \rangle : | \} \langle \text{törzs} \rangle$   
 $\langle \text{törzs} \rangle ::= \{ \text{READ} | \text{WRITE} \} \langle \text{azonosító} \rangle |$   
 $\text{LET } \langle \text{azonosító} \rangle = \langle \text{kifejezés} \rangle |$   
 $\text{GOTO } \langle \text{címke} \rangle \{ \text{IF } \langle \text{kifejezés} \rangle \} \}$   
 $\langle \text{azonosító} \rangle ::= \langle \text{betű} \rangle \{ \langle \text{számjegy} \rangle \}_0^1$   
 $\langle \text{betű} \rangle ::= X | Y | Z$   
 $\langle \text{számjegy} \rangle ::= 0 | 1 | \dots | 9$ 
```

Az EP programozási nyelv leírása EBNF-fel/2

$\langle \text{kifejezés} \rangle ::= \{ + | - | \} \langle \text{előjeltelen kifejezés} \rangle$
 $\langle \text{előjeltelen kifejezés} \rangle ::= \langle \text{tag} \rangle @ \{ \{ + | - \} \langle \text{tag} \rangle \}$
 $\langle \text{tag} \rangle ::= \langle \text{tényező} \rangle @ \{ \{ * | / \} \langle \text{tényező} \rangle \}$
 $\langle \text{tényező} \rangle ::= \langle \text{azonosító} \rangle | \langle \text{szám} \rangle | (\langle \text{kifejezés} \rangle)$
 $\langle \text{címké} \rangle ::= \langle \text{szám} \rangle$
 $\langle \text{szám} \rangle ::= \langle \text{számjegy} \rangle \{ \langle \text{számjegy} \rangle \}_0^1$

Fehér szóközök, koci vissza/soremelés karakterek.

IF: ha a kifejezés ≥ 0 , ugrik.

Példa EP-vel megoldható feladatra

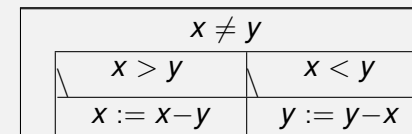
Legnagyobb közös osztó

Írjunk olyan EP programot, mely beolvas 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:



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;
```

BNF használata adatszerkezetek leírására

Bináris fák, listák

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

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

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

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

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

Kétszintű nyelvtanok

W nyelvtan

Példa:

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

Metaszabály

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

Hiperszabály

Kétszintű (E)BNF

van Wijngaarden nyelvtan (W nyelvtan)

ALGOL 68

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

Szintaxis gráfok

Definíció

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

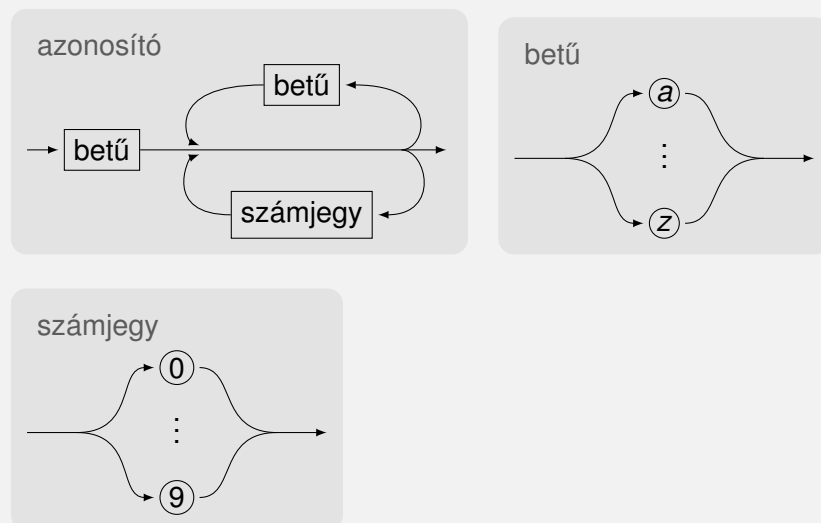
A gráf irányított, összefüggő, egy bemeneti és egy kimeneti éllel. Az élek címkézetlenek, az szögpontok vagy címkézetlenek, vagy □-be, vagy ○-be írt címkékkel ellátottak.

- A fogalmak □ címkéjű csomópontokba,
- a terminálisok (vagy terminális sorozatok) ○ címkéjű csomópontokba kerülnek,
- a gráf neve a definiálandó fogalom.

A gráf a következő értelemben írja le a definiálandó fogalmat: Tekintsünk egy irányított utat a bemeneti éltől a kimeneti élíg. Az út mentén a címkézett csúcsoknak megfelelő fogalmak és terminálisok meghatároznak egy sorozatot. A definiálandó fogalmat az összes ilyen út által meghatározott sorozatok összessége írja le.

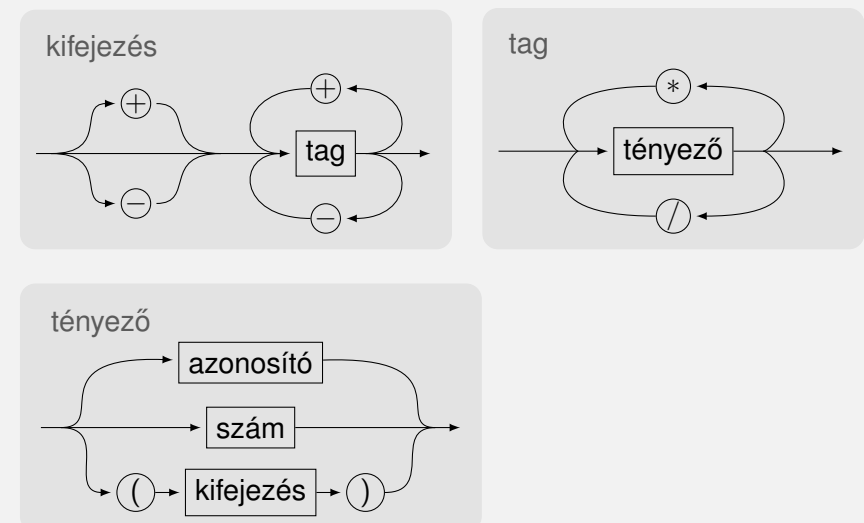
Szintaxis gráfok

Példa: Azonosító



Szintaxis gráfok

Példa: EP nyelv, kifejezés



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