

Formális nyelvek és fordítóprogramok

0. Bevezető

Bodó Zalán

Babeş–Bolyai Tudományegyetem
Matematika és Informatika Kar
Magyar Matematika és Informatika Intézet



- ▶ átírási szabályok + nemterminális szimbólumok + terminális szimbólumok

$G = (N, T, P, S)$, ahol

- ▶ N – nemterminális szimbólumok
- ▶ T – terminális szimbólumok
- ▶ P – átírási szabályok
- ▶ S – kezdőszimbólum

$Sentence \rightarrow NP + VP$

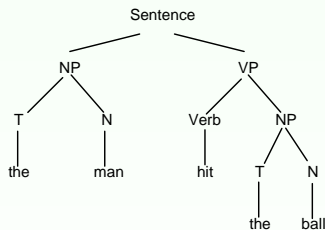
$NP \rightarrow T + N$

$VP \rightarrow Verb + NP$

$T \rightarrow the$

$N \rightarrow man, ball, etc.$

$Verb \rightarrow hit, took, etc.$



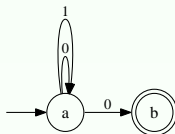
Véges automaták

- ▶ irányított gráfok + speciális csomópontok + átmenetek/átmenetszimbólumok

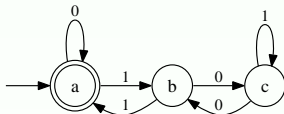
$A = \{Q, \Sigma, E, I, F\}$, ahol

- ▶ Q – állapotok halmaza
- ▶ Σ – bemeneti ábécé
- ▶ E – állapotátmenetek
- ▶ I – kezdőállapot(ok)
- ▶ F – végállapotok

páros bináris számok:



3-mal osztható bináris számok:



Generatív grammatikák, nyelvek

Véges automaták

Véremautomaták

Fordítóprogramok

(Jegy kiszámítása)

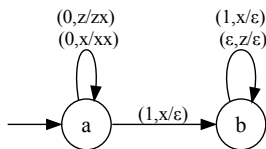
Az előadások tartalma

Ajánlott bibliográfia

Veremautomaták

$$V = (Q, \Sigma, W, E, q_0, z_0, F)$$

- ▶ Q az **állapotok** véges és nem üres halmaza
- ▶ Σ a **bemeneti ábécé**
- ▶ W a **veremábécé**
- ▶ $E \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times W \times W^* \times Q$ az **átmenetek** vagy **élek** halmaza
- ▶ $q_0 \in Q$ a **kezdőállapot**
- ▶ $z_0 \in W$ a **verem kezdőszimbóluma**
- ▶ $F \subseteq Q$ a **végállapotok** halmaza



A veremautomata által **üres veremmel** felismert nyelv:
 $\{0^n 1^n \mid n \geq 1\}$.

Fordítóprogramok

- ▶ lexikális elemzés
- ▶ szintaktikai elemzés
- ▶ szemantikai elemzés
- ▶ kódgenerálás
- ▶ kódoptimalizálás

```
int _a;
_a = 0;
while (_a < 10) {
    print _a;
    _a = _a+1;
}
```

⇒

```

;;;;;;;;;;;;;;;;;;
; PROLOGUE
;;;;;;;;;;;;;;;;;;
#include "_io_win.asm"
section .text
;;;;;;;;;;;;;;;;;;
    global _main
_main:
;;;;;;;;;;;;;;;;;;
    mov eax, 0
    mov [_a], eax
while1:
    mov eax, [_a]
    push eax
    mov eax, 10
    pop ebx
    cmp ebx, eax
    jl near yes1
    push 0
    jmp near end1
yes1:
    push 1
end1:
    pop eax
    cmp eax, 0
    je near endwhile1
    mov eax, [_a]
    push eax
    call print_int
    add esp, 4
    mov eax, [_a]
    push eax
    mov eax, 1
    pop ebx
    add eax, ebx
    mov [_a], eax
    jmp near while1
endwhile1:
;;;;;;;;;;;;;;;;;;
; EPILOGUE
;;;;;;;;;;;;;;;;;;
    push 1
    call _exit
;;;;;;;;;;;;;;;;;;
section .data
;;;;;;;;;;;;;;;;;;
    _a dd 0
section .bss
;;;;;;;;;;;;;;;;;;

```

Jegy kiszámítása

JEGY =

- ▶ **40%** laborjegy (*el kell érni minimálisan az 5-öst*)
(7 jelenlét a 14-ből kötelező; ha ez nem teljesül, a laborfeladatok pontszáma nullázódik)
- ▶ **15%** szemináriumi jegy
(4 jelenlét a 7-ből kötelező; ha ez nem teljesül, a szemináriumi tevékenység pontszáma nullázódik)
- ▶ **45%** írásbeli jegy (*el kell érni minimálisan az 5-öst*)

A végső jegynek is minimálisan 5-ösnek kell lennie ahhoz, hogy átmenőnek minősüljön.

- 1–2. Nyelvek és nyelvtanok
- 3–6. Véges automaták és reguláris nyelvek
 7. Veremautomaták és környezetfüggetlen nyelvek
 8. Fordítóprogramok. Lexikális elemzés
 9. A `flex` és a `bison`
- 10–12. Szintaktikai elemzések
13. Szimbólumtábla, szemantikai elemzés, kódgenerálás, kódoptimalizálás
vagy
ISO/RFC/ECMA, C++, (Java,) BNF, Turing
14. Összefoglalás, ismétlés

Ajánlott bibliográfia

1. Kása Zoltán. Automaták és formális nyelvek. In A. Iványi (szerk.). *Informatikai algoritmusok 2*, Eötvös ELTE Kiadó, Budapest, 2005, 19. fejezet
www.cs.ubbcluj.ro/~kasa/formal/Automatak.pdf
2. Kása Zoltán. *Formális nyelvek és automaták*. Farkas Gyula Egyesület, 2004.
3. Bach Iván. *Formális nyelvek*. Typotex, Budapest, 2002.
mek.niif.hu/05000/05099/05099.pdf
4. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. *Compilers. Principles, Techniques, & Tools*. Pearson Education, 2006.
5. Csörnyei Zoltán, Kása Zoltán. **Formális nyelvek és fordítóprogramok, Kolozsvári Egyetemi Kiadó, 2007.**
www.ms.sapientia.ro/~kasa/CsornyeiKasa.zip
6. Csörnyei Zoltán. *Fordítási algoritmusok*, Erdélyi Tankönyvtanács, Kolozsvár, 2000.
7. Csörnyei Zoltán. *Fordítóprogramok*. Typotex, Budapest, 2006.
8. Charles Donnelly, Richard Stallman. Bison. *The YACC-compatible Parser Generator*, 2012.
www.gnu.org/software/bison/manual/bison.pdf
9. Vern Paxson. *Flex, version 2.5. A fast scanner generator*, 1995.
www-2.dc.uba.ar/materias/tl/dload/flex.pdf
flex.sourceforge.net/manual/

10. Anthony A. Aaby. *Compiler Construction using Flex and Bison*, 2005. yilmazkilicaslan.trakya.edu.tr/teaching/compiler.pdf
11. **Bodó Zalán. Fordítóprogramok szerkesztése Flex és Bison segítségével. Erdélyi Múzeum-Egyesület, Kolozsvár, 2014.** <http://eda.eme.ro/handle/10598/28426>
12. Emden Gansner, Eleftherios Koutsofios, Stephen North. *Drawing graphs with dot*, 2006. (a Graphviz csomagban is megtalálható) www.graphviz.org/Documentation/dotguide.pdf