reg. languages	FA	reg. grammar	reg. expression
determ. cf. languages	DPDA		
cf. languages	PDA	cf. grammar	
cs. languages	LBA	cs. grammar	
re. languages	TM	unrestr. grammar	

- Deadline huiswerkopgave 1: dinsdag 7 oktober, 23.59 uur
- Huiswerkopgave 2: eind deze week / begin volgende week bekend

$Pal = \{x \in \{a, b\}^* \mid x^r = x\}$ (reverse)
In canonical (shortlex) order:
$\Lambda, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, abba, \ldots$
Recursive definition:

Example

– $\Lambda, a, b \in Pal$
– for every $x \in Pal$, also $axa, bxb \in Pal$

$S \to \Lambda \mid a \mid b$
$S \to aSa$
$S \to bSb$

$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aababaa$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa and bSb are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa, bSb, aSb and bSa are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb \mid aSb \mid bSa$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots$
[M] E 4.3

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every S in $NonPal$, aSa, bSb, aSb and bSa are in $NonPal$
– for every $A \in \{a, b\}^*$, aAb and bAa are elements of $NonPal$

$S \to aAb \mid bAa \mid aSa \mid bSb \mid aSb \mid bSa$
$A \to \Lambda \mid aA \mid bA$

$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbAaba \Rightarrow abbbAaba \Rightarrow abbbaAaba \Rightarrow abbbaaba$
$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbSaba \Rightarrow abbbAaaba \Rightarrow abbbaaba$
[M] E 4.3

Balanced

$\mathit{Balanced} \subseteq \{(,)\}^*$: $\Lambda, (), (()), ()(), ((())), (())(), \ldots$
Recursive definition:

Example

– $\Lambda \in \mathit{Balanced}$
– for every $x, y \in \mathit{Balanced}$, also $xy \in \mathit{Balanced}$
– for every $x \in \mathit{Balanced}$, also $(x) \in \mathit{Balanced}$

[M] E 1.19

Balanced $\subseteq \{(,)\}^*$: $\Lambda, (), (()), ()(), ((())), (())(), \ldots$
Recursive definition:

Example

– $\Lambda \in Balanced$
– for every $x, y \in Balanced$, also $xy \in Balanced$
– for every $x \in Balanced$, also $(x) \in Balanced$

$S \to \Lambda \mid SS \mid (S)$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()((S)) \Rightarrow ()(())$

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

[M] E 1.19

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

$S \to a \mid S + S \mid S * S \mid (S)$
derivation(s) for $a + (a * a)$ and $a + a * a \ldots$

[M] E 4.2

$Expr \subseteq \{a, +, *, (,)\}^*$
Recursive definition:

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, also $(x) \in Expr$

$S \to a \mid S + S \mid S * S \mid (S)$

$S \Rightarrow S + S \Rightarrow S + S * S$
$S \Rightarrow S * S \Rightarrow S + S * S$

ambiguity

[M] E 4.2

alphabet $\{\ 1, 2, 5, =\ \}$

$\{\ x{=}y \mid x \in \{1, 2\}^*, y \in \{5\}^*, n_1(x) + 2n_2(x) = 5n_5(y)\ \}$

$n_\sigma(x)$ number of σ occurrences in x

212=5   22222=55   121221221222=5555

The problem with most solutions is that when read from left to right the initial string over $\{1, 2\}$ cannot always be chopped into parts with exact value 5, without chopping the symbol 2.
The solution is like a finite automaton, which reads $1, 2$ and 'saves' the values until the value 5 is reached, then we write a 5 to the right.

alphabet $\{ 1, 2, 5, = \}$

$\{ x{=}y \mid x \in \{1,2\}^*, y \in \{5\}^*, n_1(x) + 2n_2(x) = 5n_5(y) \}$

$n_\sigma(x)$ number of σ occurrences in x

212=5 22222=55 121221221222=5555

variables S_i, $0 \le i \le 4$

axiom S_0

productions

$S_0 \to 1S_1 \mid 2S_2$
$S_1 \to 1S_2 \mid 2S_3$
$S_2 \to 1S_3 \mid 2S_4$
$S_3 \to 1S_4 \mid 2S_05$
$S_4 \to 1S_05 \mid 2S_15$
$S_0 \to =$

Definition

context-free grammar (CFG) 4-tuple $G = (V, \Sigma, S, P)$

– V alphabet *variables* / *nonterminals*

– Σ alphabet *terminals* disjoint $V \cap \Sigma = \varnothing$

– $S \in V$ *axiom*, *start symbol*

– P finite set rules, *productions*

of the form $A \to \alpha$, $A \in V$, $\alpha \in (V \cup \Sigma)^*$

derivation step $\alpha = \alpha_1 A \alpha_2 \Rightarrow_G \alpha_1 \gamma \alpha_2 = \beta$ for $A \to \gamma \in P$

Definition

language generated by G

$L(G) = \{ x \in \Sigma^* \mid S \Rightarrow_G^* x \}$

[M] Def 4.6 & 4.7

NonPal, its grammar components

$$S \to aAb \mid bAa \mid aSa \mid bSb$$
$$A \to \Lambda \mid aA \mid bA$$

variables $\quad V = \{ S, A \}$
terminals $\quad \Sigma = \{ a, b \}$
axiom $\quad S$
productions
$P = \{A \to \Lambda, A \to aA, A \to bA, S \to aAb, S \to bAa, S \to aSa, S \to bSb\}$

$\Rightarrow_G^*$ is the *transitive and reflexive closure* of $\Rightarrow_G$

zero, one or more steps

general case $\quad \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = \beta$

$\alpha \Rightarrow_G^* \beta$ iff there are strings $\alpha_0, \alpha_1, \ldots, \alpha_n$ such that
- $\alpha_0 = \alpha$
- $\alpha_n = \beta$
- $\alpha_i \Rightarrow \alpha_{i+1}$ $\quad$ for $0 \leq i < n$.

special case $\quad n = 0 \quad \alpha = \alpha_0 = \beta$

Variables can be rewritten regardless of context

Lemma

If $u_1 \Rightarrow^* v_1$ *and* $u_2 \Rightarrow^* v_2$*, then* $u_1 u_2 \Rightarrow^* v_1 v_2$*.*

Lemma

If $u \Rightarrow^* v_1 v v_2$ *and* $v \Rightarrow^* w$*, then* $u \Rightarrow^* v_1 w v_2$*.*

Lemma

If $u \Rightarrow^* v$ *and* $u = u_1 u_2$*,*
then $v = v_1 v_2$ *such that* $u_1 \Rightarrow^* v_1$ *and* $u_2 \Rightarrow^* v_2$*.*

$AEqB = \{ \ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \ \}$

$aaabbb, ababab, aababb, \ldots$

[M] E 4.8

From lecture 5:

– Even number of both a and b

two letters together

aa and bb keep both numbers even [odd]

ab and ba switch between even and odd, for both numbers

$(\ aa + bb + (ab + ba)(aa + bb)^*(ab + ba) \ )^*$

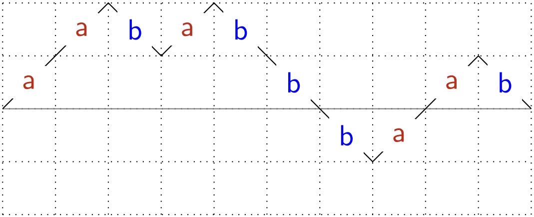
[M] E 3.4

$AEqB = \{ \, x \in \{a, b\}^* \mid n_a(x) = n_b(x) \, \}$

$aaabbb, ababab, aababb, \ldots$

[M] E 4.8

$$AEqB = \{\ x \in \{a,b\}^* \mid n_a(x) = n_b(x)\ \}$$



$AEqB = \{ \ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \ \}$

$aaabbb, ababab, aababb, \ldots$

$$\begin{aligned} S &\to \Lambda \mid aB \mid bA \\ A &\to aS \mid bAA \\ B &\to bS \mid aBB \end{aligned}$$

A generates $n_a(x) = n_b(x) + 1$
B generates $n_a(x) + 1 = n_b(x)$

$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow \ldots$ (different options)

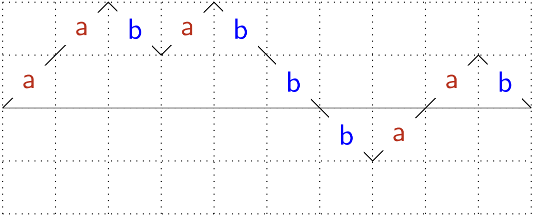
(1) $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$
(2) $\ldots$ (ambiguous, later)

[M] E 4.8

ABOVE

When a string has multiple variables, like $aabSB$ in the above example, then we are not forced to rewrite the first variable, we can as well rewrite another one.
Thus we can do $aab\underline{S}B \Rightarrow aabB$, but also $aabS\underline{B} \Rightarrow aabSaBB$, for instance.

$AEqB = \{ \; x \in \{a,b\}^* \mid n_a(x) = n_b(x) \; \}$



$S \to \Lambda \mid aSb \mid bSa \mid SS$

$S \Rightarrow SS \Rightarrow a_1 S b_6 S \Rightarrow a_1 a_2 S b_3 S b_6 S \Rightarrow \ldots$

$S \Rightarrow a_1 S b_{10} \Rightarrow \ldots$

[M] Exercise 1.66

$L_1 = \{ a^i b^j c^k \mid i = j + k \}$ $\quad aaa\,b\,cc$

$L_1 = \{ \; a^i b^j c^k \mid i = j + k \; \}$ $\quad aaa\,b\,cc$

generate as $\quad a^{k+j}\; b^j\; c^k = \underbrace{a^k\, \underbrace{a^j\; b^j}\, c^k}$

$S \to aSc \mid T$
$T \to aTb \mid \Lambda$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$

$L_1 = \{ a^i b^j c^k \mid i = j + k \}$ $\quad aaa\, b\, cc$

generate as $\quad a^{k+j}\ b^j\ c^k = \underbrace{a^k\ \underbrace{a^j\ b^j}\ c^k}$

$$\begin{aligned} S &\to aSc \mid T \\ T &\to aTb \mid \Lambda \end{aligned}$$

$$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$$

$L_2 = \{ a^i b^j c^k \mid j = i + k \}$ $\quad a\, bbb\, cc$

$L_1 = \{ \ a^i b^j c^k \mid i = j + k \ \}$ $\quad aaa\,b\,cc$

generate as $\quad a^{k+j}\ b^j\ c^k = \underbrace{a^k\ \underbrace{a^j\ b^j}\ c^k}$

$S \to aSc \mid T$
$T \to aTb \mid \Lambda$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$

$L_2 = \{ \ a^i b^j c^k \mid j = i + k \ \}$ $\quad a\,bbb\,cc$

generate as $\quad a^i\ b^{i+k}\ c^k = \underbrace{a^i\ b^i}\ \underbrace{b^k\ c^k}$

$S \to XY$ $\quad$ (concatenate)
$X \to aXb \mid \Lambda$
$Y \to bYc \mid \Lambda$

$S \Rightarrow \underline{X}\,Y \Rightarrow a\underline{X}b\,Y \Rightarrow ab\,\underline{Y} \Rightarrow ab\,b\underline{Y}c \Rightarrow ab\,bb\underline{Y}cc \Rightarrow abbbcc$
$S \Rightarrow X\,\underline{Y} \Rightarrow \underline{X}\,bYc \Rightarrow aXb\,b\underline{Y}c \Rightarrow a\underline{X}b\,bbYcc \Rightarrow ab\,bb\underline{Y}cc \Rightarrow abbbcc$

(a priori there is no prescribed order rewriting X or Y)

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

[M] Thm 4.9

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, L_1L_2 and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.
[M] Thm 4.9

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.

Construction

$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P)$, new axiom S

– $P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$ $\quad L(G) = L(G_1) \cup L(G_2)$

– $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$ $\quad L(G) = L(G_1) L(G_2)$

$G = (V_1 \cup \{S\}, \Sigma, S, P)$, new axiom S

– $P = P_1 \cup \{S \to SS_1, S \to \Lambda\}$ $\quad L(G) = L(G_1)^*$

Proof. . .

[M] Thm 4.9

$$L_0 = \{ a^i b^j c^k \mid j = i + k \} = \{ a^i b^{i+k} c^k \mid i, k \geq 0 \}$$
$$= \{ \underbrace{a^i b^i}\underbrace{b^k c^k} \mid i, k \geq 0 \}$$

$$S_0 \to XY \quad X \to aXb \mid \Lambda \quad Y \to bYc \mid \Lambda$$

$$L = \{ a^i b^j c^k \mid j \neq i + k \}$$

[M] E 4.10

$$L_0 = \{\ a^i b^j c^k \mid j = i + k\ \} = \{\ a^i b^{i+k} c^k \mid i, k \geq 0\ \}$$
$$= \{\ \underbrace{a^i b^i}\,\underbrace{b^k c^k} \mid i, k \geq 0\ \}$$

$$S_0 \to XY \qquad X \to aXb \mid \Lambda \qquad Y \to bYc \mid \Lambda$$

$$L = \{\ a^i b^j c^k \mid j \neq i + k\ \} = L_1 \cup L_2$$
$$S \to S_1 \mid S_2$$

$$L_1 = \{\ a^i b^j c^k \mid j > i + k\ \}$$
$$L_2 = \{\ a^i b^j c^k \mid j < i + k\ \}$$

[M] E 4.10

$$L_0 = \{ a^i b^j c^k \mid j = i + k \} = \{ a^i b^{i+k} c^k \mid i, k \geq 0 \}$$
$$= \{ \underbrace{a^i b^i}\underbrace{b^k c^k} \mid i, k \geq 0 \}$$

$S_0 \to XY \quad X \to aXb \mid \Lambda \quad Y \to bYc \mid \Lambda$

$L = \{ a^i b^j c^k \mid j \neq i + k \} = L_1 \cup L_2$

$S \to S_1 \mid S_2$

$L_1 = \{ a^i b^j c^k \mid j > i + k \}$

$S_1 \to X_1 b Y_1$
$X_1 \to a X_1 b \mid X_1 b \mid \Lambda$
$Y_1 \to b Y_1 c \mid b Y_1 \mid \Lambda$

$L_2 = \{ a^i b^j c^k \mid j < i + k \}$

[M] E 4.10

$$L_0 = \{\ a^i b^j c^k \mid j = i + k\ \} = \{\ a^i b^{i+k} c^k \mid i, k \geq 0\ \}$$
$$= \{\ \underbrace{a^i b^i}\underbrace{b^k c^k} \mid i, k \geq 0\ \}$$

$S_0 \to XY \qquad X \to aXb \mid \Lambda \qquad Y \to bYc \mid \Lambda$

$L = \{\ a^i b^j c^k \mid j \neq i + k\ \} = L_1 \cup L_2$

$S \to S_1 \mid S_2$

$L_1 = \{\ a^i b^j c^k \mid j > i + k\ \}$

$S_1 \to X_1 b Y_1$
$X_1 \to aX_1 b \mid X_1 b \mid \Lambda$
$Y_1 \to bY_1 c \mid bY_1 \mid \Lambda$

$L_2 = \{\ a^i b^j c^k \mid j < i + k\ \}$

$S_2 \to aX_2 Y_2 \mid X_2 Y_2 c$
$X_2 \to aX_2 b \mid aX_2 \mid \Lambda$
$Y_2 \to bY_2 c \mid Y_2 c \mid \Lambda$

[M] E 4.10

ABOVE

The solution in the book for $j \neq i + k$ is a bit complex. We have made it shorter here.

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.

Construction

$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P)$, new axiom S

– $P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$ $\quad L(G) = L(G_1) \cup L(G_2)$

– $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$ $\quad L(G) = L(G_1) L(G_2)$

$G = (V_1 \cup \{S\}, \Sigma, S, P)$, new axiom S

– $P = P_1 \cup \{S \to S S_1, S \to \Lambda\}$ $\quad L(G) = L(G_1)^*$

Proof. . .

[M] Thm 4.9

Regular languages are closed under
- Boolean operations (complement, union, intersection, minus)
- Regular operations (union, concatenation, star)
- Reverse (mirror)
- [inverse] Homomorphism

Fact, proof follows $\hookrightarrow$later

Theorem

the languages
– $AnBnCn = \{ \, a^n b^n c^n \mid n \geq 0 \, \}$ *and*
– $XX = \{ xx \mid x \in \{a, b\}^* \, \}$
are not context-free

[M] E 6.3, E 6.4

AnBnCn is the intersection of two context-free languages
[M] E 6.10

The complement of both *AnBnCn* and *XX* is context-free.
[M] E 6.11
Hence, CFL is not closed under intersection, complement

$S \to S_1 \mid S_2$ union
$S \to S_1 S_2$ concatenation
$S \to SS_1 \mid \Lambda$ star

CFG for $\varnothing$...
CFG for $\{\sigma\}$...

Example

$L = bba(ab)^* + (ab + ba^*b)^*ba$

[M] E 4.11

$S \to S_1 \mid S_2$ union
$S \to S_1 S_2$ concatenation
$S \to S S_1 \mid \Lambda$ star

Example

$L = bba(ab)^* + (ab + ba^* b)^* ba$
$S \to S_1 \mid S_2$
$S_1 \to S_1 ab \mid bba$
$S_2 \to T S_2 \mid ba \quad T \to ab \mid bUb \quad U \to aU \mid \Lambda$

[M] E 4.11

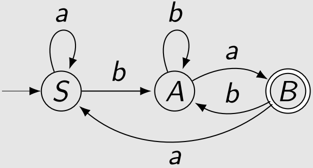
ABOVE

We have seen constructions to apply the regular operations (union, concatenation and star) to context-free grammars. These we can now use to build CFG for regular expressions.

There is a better way to build CFG for regular languages. Use finite automata, and simulate these using a very simple type of context-free grammar. These simple grammars are called regular.

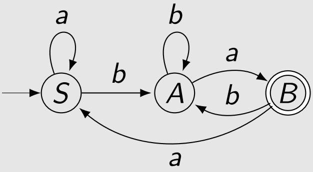
systematic approach

Example



systematic approach

Example



axiom S	initial state
$S \to bA \mid aS$	transitions
$A \to bA \mid aB$	
$B \to bA \mid aS$	
$B \to \Lambda$	accepting state

path / derivation for *bbaaba*. . .

Definition

regular grammar (or *right-linear grammar*)
productions are of the form
– $A \to \sigma B$ variables A, B, terminal σ
– $A \to \Lambda$ variable A

Special type of context-free grammar

Theorem

A language L is regular,
if and only if there is a regular grammar generating L.

Proof. . .

[M] Def 4.13, Thm 4.14