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

$aaabbb, ababab, aababb, \ldots$

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

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)

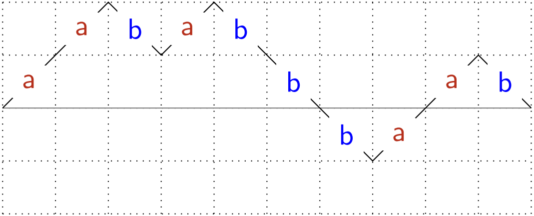
[M] E 4.8

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

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

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

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

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

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

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

$i = j + k$ vs $j = i + k$

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

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

$$S \to XY \qquad \text{(concatenate)}$$
$$X \to aXb \mid \Lambda$$
$$Y \to bYc \mid \Lambda$$

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

(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 $L_1 \cup L_2$, $L_1 L_2$ 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
- $P = P_1 \cup \{S \to S S_1, S \to \Lambda\}$ $\quad L(G) = L(G_1)^*$

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

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

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

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

$S \to S_1 \mid S_2$

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

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

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

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

$G = (V_1 \cup \{S\}, \Sigma, S, P)$, new axiom S
– $P = P_1 \cup \{S \to S S_1, S \to \Lambda\}$ $L(G) = L(G_1)^*$

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

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
$S \to SS_1 \mid \Lambda$	star

Example

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

$S \to S_1 \mid S_2$

$S_1 \to S_1 ab \mid bba$

$S_2 \to TS_2 \mid ba \qquad T \to ab \mid bUb \qquad U \to aU \mid \Lambda$

[M] E 4.11

Note

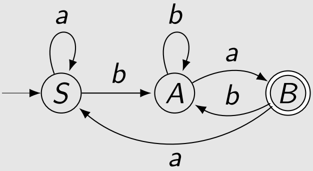
ABOVE

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.

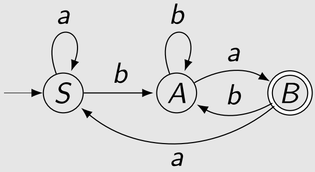
systematic approach

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

4.4 Derivation trees and ambiguity

A derivation...

$S \to a \mid S + S \mid S * S \mid (S) \qquad \Sigma = \{a, +, *, (,)\}$

$$S \Rightarrow S + \underline{S} \Rightarrow S + (\underline{S}) \Rightarrow S + (\underline{S} * S) \Rightarrow$$
$$\underline{S} + (a * S) \Rightarrow a + (a * \underline{S}) \Rightarrow a + (a * a)$$

[M] E 4.2, Fig 4.15

Definition

A derivation in a context-free grammar is a *leftmost* derivation, if at each step, a production is applied to the leftmost variable-occurrence in the current string.
A *rightmost* derivation is defined similarly.

[M] D 4.16

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

The derivation step is *leftmost* iff $\alpha_1 \in \Sigma^*$

We write $\alpha \overset{\ell}{\Rightarrow} \beta$

$$S \to a \mid S + S \mid S * S \mid (S) \qquad \Sigma = \{a, +, *, (,)\}$$

$$S \Rightarrow S + \underline{S} \Rightarrow S + (\underline{S}) \Rightarrow S + (\underline{S} * S) \Rightarrow$$
$$\underline{S} + (a * S) \Rightarrow a + (a * \underline{S}) \Rightarrow a + (a * a)$$

Derivation tree...

[M] E 4.2, Fig 4.15