Uitslagen huiswerkopgave 1...

From lecture 7:

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

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

Proof. . .

[M] Thm 4.9

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

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

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

[M] E 4.10

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

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

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

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

[M] E 4.10

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

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

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

$S \to S_1 \mid S_2$

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

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

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

[M] E 4.10

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

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

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

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

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

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

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

[M] E 4.10

ABOVE

De uitwerking uit het boek is wat te ingewikkeld, dat hebben we hier wat ingekort.

From lecture 7:
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

Hence, CFL is closed onder union, concatenation, star

Regular languages are closed under

– Boolean operations (complement, union, intersection, minus)
– Regular operations (union, concatenation, star)
– Reverse (mirror)
– [inverse] Homomorphism

Fact, proof follows ↪later

Theorem

the languages
– $AnBnCn = \{ a^n b^n c^n \mid n \geqslant 0 \}$ *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

$$
\begin{array}{ll}
S \to S_1 \mid S_2 & \text{union} \\
S \to S_1 S_2 & \text{concatenation} \\
S \to SS_1 \mid \Lambda & \text{star}
\end{array}
$$

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
$S \to SS_1 \mid \Lambda$ star

Example

$L = bba(ab)^* + (ab + ba^*b)^*ba$
$S \to S_1 \mid S_2$
$S_1 \to S_1 ab \mid bba$
$S_2 \to TS_2 \mid ba \quad T \to ab \mid bUb \quad U \to aU \mid \Lambda$

[M] E 4.11

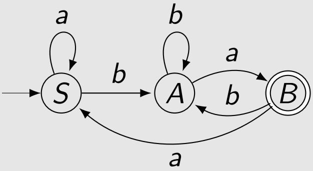
ABOVE

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.

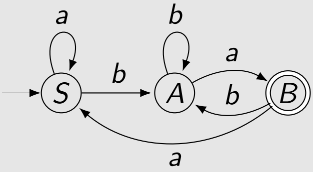
systematic approach

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

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 $\alpha_1 \in \Sigma^*$

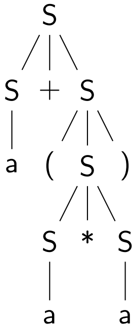
We write $\alpha \overset{\ell}{\Rightarrow} \beta$

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

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

Derivation tree...

[M] E 4.2, Fig 4.15

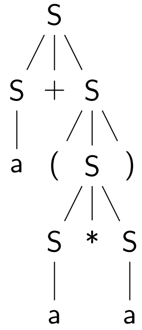


[M] E 4.2, Fig 4.15

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

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

Leftmost derivation. . .



[M] E 4.2, Fig 4.15

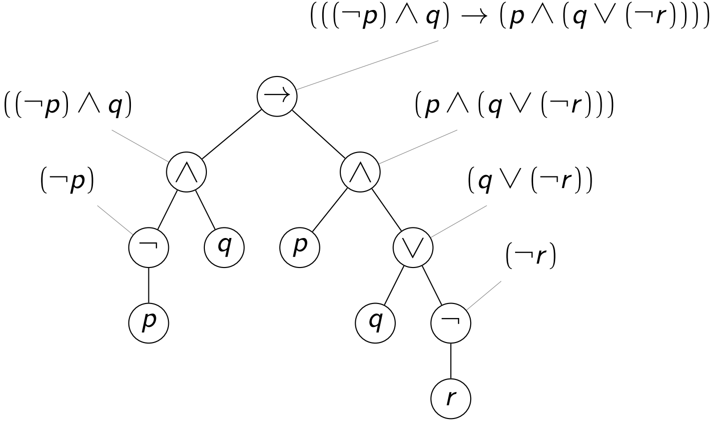
$$S \rightarrow a \mid S + S \mid S * S \mid (S) \qquad \Sigma = \{a, +, *, (,)\}$$

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

Leftmost derivation:

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

$\psi ::= p \mid (\neg\psi) \mid (\psi \wedge \psi) \mid (\psi \vee \psi) \mid (\psi \rightarrow \psi)$

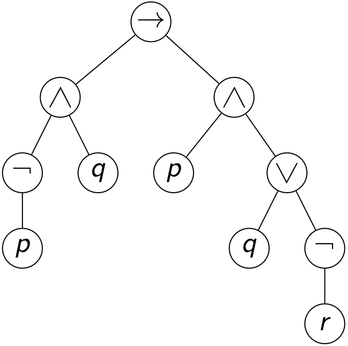


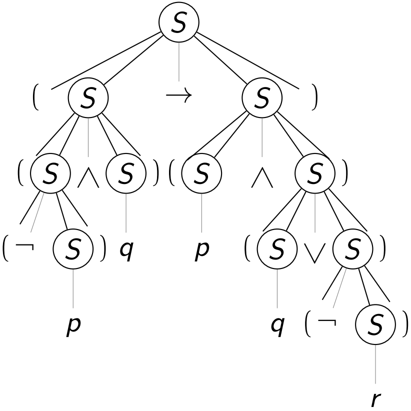
[H&R] Fig 1.3

Well-formed formula

$S ::= p \mid q \mid r \mid (\neg S) \mid (S \wedge S) \mid (S \vee S) \mid (S \to S)$

parse tree vs. derivation tree[2]



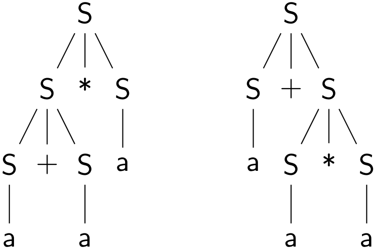


[2]with all brackets explicit

Definition

A context-free grammar G is *ambiguous*, if for at least one $x \in L(G)$, x has more than one derivation tree.

Otherwise: *unambiguous* [M] D 4.18



$\Sigma = \{a, +, *, (,)\}$

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

$a + a * a$

leftmost derivation $\longleftrightarrow$ derivation tree

Theorem

If G is a context-free grammar, then for every $x \in L(G)$, these three statements are equivalent:

1. *x has more than one derivation tree*
2. *x has more than one leftmost derivation*
3. *x has more than one rightmost derivation*

Proof...

[M] Thm 4.17

Ambiguity

leftmost derivation $\longleftrightarrow$ derivation tree

Theorem

If G is a context-free grammar, then for every $x \in L(G)$, these three statements are equivalent:

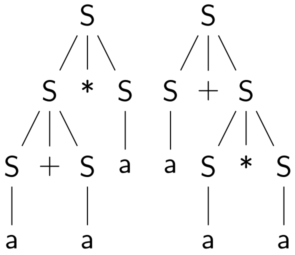
1. *x has more than one derivation tree*
2. *x has more than one leftmost derivation*
3. *x has more than one rightmost derivation*

[M] Thm 4.17

Definition

A context-free grammar G is *ambiguous*, if for at least one $x \in L(G)$, x has more than one derivation tree (or, equivalently, more than one leftmost derivation).

Otherwise: *unambiguous* [M] D 4.18



$\Sigma = \{a, +, *, (,)\}$

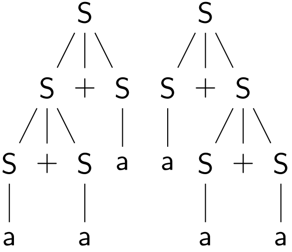
$S \rightarrow a \mid S + S \mid S * S \mid (S)$

$a + a * a$

$S \overset{\ell}{\Rightarrow} \underline{S} * S \overset{\ell}{\Rightarrow} S + S * S \overset{\ell}{\Rightarrow} a + S * S \overset{\ell}{\Rightarrow} a + a * S \overset{\ell}{\Rightarrow} a + a * a$

$S \overset{\ell}{\Rightarrow} \underline{S}{+}S \overset{\ell}{\Rightarrow} a{+}S \overset{\ell}{\Rightarrow} a{+}S{*}S \overset{\ell}{\Rightarrow} a{+}a{*}S \overset{\ell}{\Rightarrow} a + a * a$

leftmost derivation $\longleftrightarrow$ derivation tree



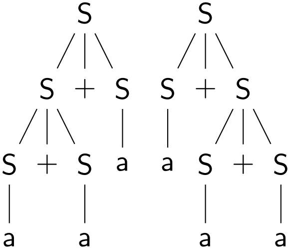
$\Sigma = \{a, +, *, (,)\}$

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

$a + a + a$

$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} S + S + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow}$
$a + a + S \overset{\ell}{\Rightarrow} a + a + a$

$S \Rightarrow S + \underline{S} \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow$
$a + a + S \Rightarrow a + a + a$



$\Sigma = \{a, +, *, (,)\}$

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

$a + a + a$

$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} S + S + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow}$
$a + a + S \overset{\ell}{\Rightarrow} a + a + a$

$S \Rightarrow S + \underline{S} \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow$
$a + a + S \Rightarrow a + a + a$

$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} a + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow}$
$a + a + S \overset{\ell}{\Rightarrow} a + a + a$

leftmost derivation $\longleftrightarrow$ derivation tree

ABOVE

This example is a little weird. In the derivation step $S+S \Rightarrow S+S+S$ we cannot really see which S has been rewritten.

Expr

ambiguous:

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

[M] E 4.20

$a + a * a$

unambiguous:

...

Expr

ambiguous:

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

[M] E 4.20

$a + a * a$

unambiguous:

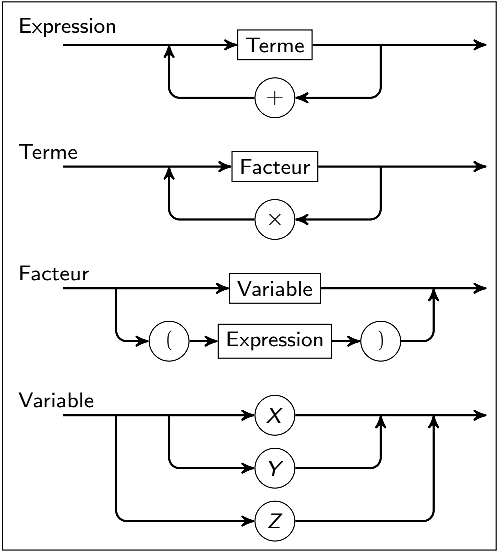
$S \to S + T \mid T$

$T \to T * F \mid F$

$F \to a \mid (S)$

[M] Thm 4.25

The proof of the unambiguity does not have to be known for the exam



http://math.et.info.free.fr/TikZ/index.html Chapitre 7

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

aaabbb, *ababab*, *aababb*, ...

$$S \to \Lambda \mid aB \mid bA$$
$$A \to aS \mid bAA \qquad A \text{ generates } n_a(x) = n_b(x) + 1$$
$$B \to bS \mid aBB \qquad B \text{ generates } n_a(x) + 1 = n_b(x)$$

Derivation for *aababb*:

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

(1) $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$
(2) $aaba\underline{B}B \Rightarrow aabab\underline{S}B \Rightarrow aabab\underline{B} \Rightarrow aababb\underline{S} \Rightarrow aababb$
(2') $aabaB\underline{B} \Rightarrow aaba\underline{B}bS \Rightarrow aababSb\underline{S} \Rightarrow aabab\underline{S}b \Rightarrow aababb$

[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.

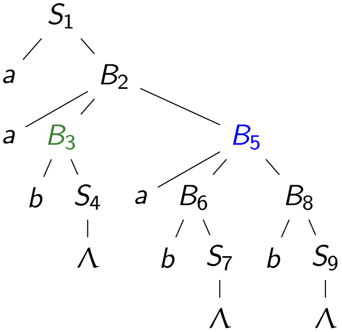
Thus we can do $aab\underline{S}B \Rightarrow aabB$, but also $aabS\underline{B} \Rightarrow aabSaBB$, for instance.

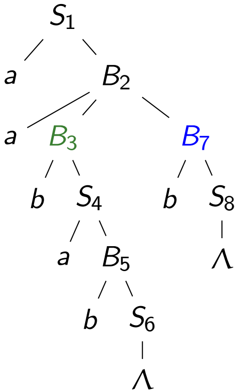
BELOW

In detail, two different derivation trees for the same string, corresponding to derivations (1) and (2,2') respectively, together with two associated leftmost derivations.

Given these two trees we conclude the grammar is ambiguous.

Derivation tree & leftmost derivations





$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$

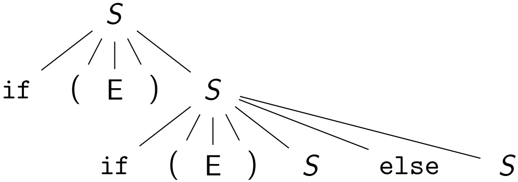
$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$

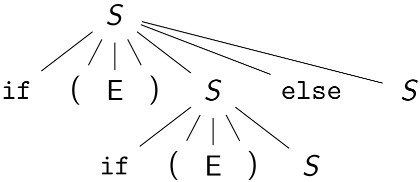
$S \rightarrow \texttt{if}\ (\ E\ )\ S \mid \texttt{if}\ (\ E\ )\ S\ \texttt{else}\ S \mid \ldots$

$\texttt{if}\ (\ E\ )\ \texttt{if}\ (\ E\ )\ S\ \texttt{else}\ S$

[M] E 4.19

$S \rightarrow \texttt{if}\ (\ E\ )\ S\ \mid\ \texttt{if}\ (\ E\ )\ S\ \texttt{else}\ S\ \mid\ \ldots$





[M] E 4.19

ambiguous:

$S \rightarrow \texttt{if}\ (\ E\ )\ S \mid \texttt{if}\ (\ E\ )\ S\ \texttt{else}\ S \mid A \mid \ \ldots$

unambiguous. . .

[M] E 4.19

ambiguous:

$S \to \texttt{if}\ (\ E\ )\ S \mid \texttt{if}\ (\ E\ )\ S\ \texttt{else}\ S \mid A \mid \ \ldots$

unambiguous:

$S \to S_1 \mid S_2$

$S_1 \to \texttt{if}\ (\ E\ )\ S_1\ \texttt{else}\ S_1 \mid A \mid \ \ldots$ (matched)

$S_2 \to \texttt{if}\ (\ E\ )\ S \mid \texttt{if}\ (\ E\ )\ S_1\ \texttt{else}\ S_2$ (open)

[M] E 4.19

Balanced

ambiguous:

$S \to SS \mid (S) \mid \Lambda$ (more or less the definition of balanced)

unambiguous:

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

[M] Exercise 4.45

Exercise.

Let G be a context-free grammar with start variable S and the following productions:

$$S \to aSbS \mid bSaS \mid \Lambda$$

a. Show that $L(G) = AEqB = \{x \in \{a,b\}^* \mid n_a(x) = n_b(x)\}$

b. Is G ambiguous? Motivate your answer.

Some cf languages are inherently ambiguous

Ambiguity is *undecidable*

[M] Theorem 9.20