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

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

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

$$L_1 = \{ a^i b^j c^k \mid j > i + k \}$$
$$S_1 \to X_1 b Y_1$$
$$X_1 \to aX_1 b \mid X_1 b \mid \Lambda$$
$$Y_1 \to bY_1 c \mid bY_1 \mid \Lambda$$

$$L_2 = \{ a^i b^j c^k \mid j < i + k \}$$
$$S_2 \to aX_2 Y_2 \mid X_2 Y_2 c$$
$$X_2 \to aX_2 b \mid aX_2 \mid \Lambda$$
$$Y_2 \to bY_2 c \mid Y_2 c \mid \Lambda$$

[M] E 4.10

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

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

$L = bba(ab)^* + (ab + ba^*b)^*ba$
$S \to S_1 \mid S_2$
$S_1 \to S_1ab \mid bba$
$S_2 \to TS_2 \mid ba \quad T \to ab \mid bUb \quad U \to aU \mid \Lambda$

[M] E 4.11

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 $\alpha_1 \in \Sigma^*$

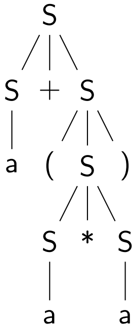
We write $\alpha \stackrel{\ell}{\Rightarrow} \beta$

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

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

Derivation tree...

[M] E 4.2, Fig 4.15

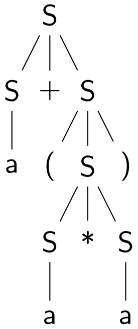


[M] E 4.2, Fig 4.15

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

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

Leftmost derivation. . .



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

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

Leftmost derivation:

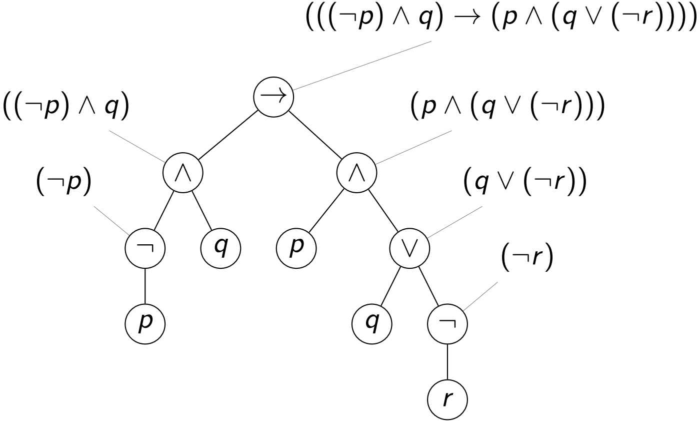
$$S \stackrel{\ell}{\Rightarrow} \underline{S} + S \stackrel{\ell}{\Rightarrow} a + S \stackrel{\ell}{\Rightarrow} a + (\underline{S}) \stackrel{\ell}{\Rightarrow} a + (\underline{S} * S) \stackrel{\ell}{\Rightarrow} a + (a * \underline{S}) \stackrel{\ell}{\Rightarrow} a + (a * a)$$

One tree $\leftrightarrow$ multiple derivations

Tree $\to$ derivation $\to$ (same) tree

[M] E 4.2, Fig 4.15

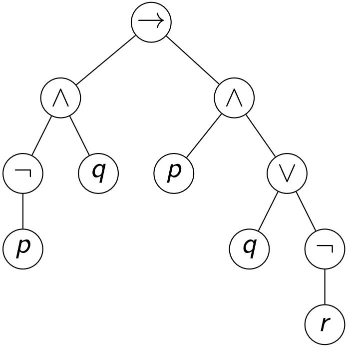
$$\psi ::= p \mid (\neg\psi) \mid (\psi \wedge \psi) \mid (\psi \vee \psi) \mid (\psi \rightarrow \psi)$$

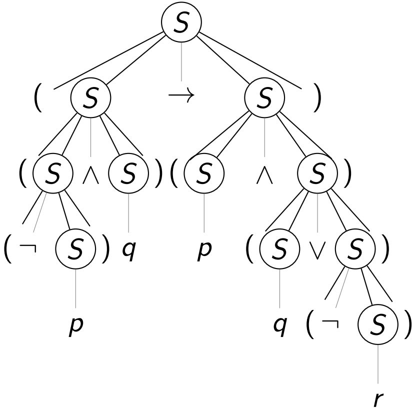


[H&R] Fig 1.3

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

parse tree vs. derivation tree[2]



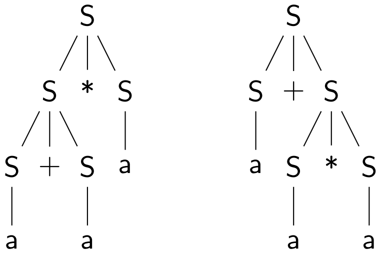


[2] with all brackets explicit

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



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

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

$a + a * a$

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*

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

[M] Thm 4.17

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:

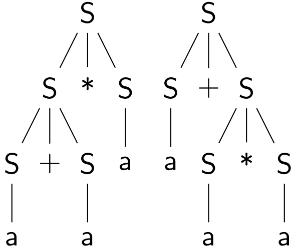
1. *x has more than one derivation tree*
2. *x has more than one leftmost derivation*
3. *x has more than one rightmost derivation*

[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



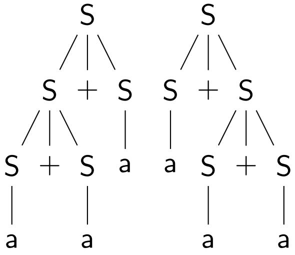
$\Sigma = \{a, +, *, (,)\}$
$S \to a \mid S + S \mid S * S \mid (S)$

$a + a * a$

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

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

leftmost derivation $\longleftrightarrow$ derivation tree



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

$a + a + a$

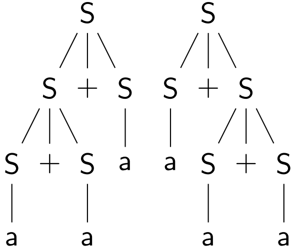
Leftmost for 1:
$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} S + S + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow} a + a + S \overset{\ell}{\Rightarrow} a + a + a$

Derivation for 2:
$S \Rightarrow S + \underline{S} \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$

Ambiguity (2)

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

$a + a + a$



Leftmost for 1:
$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} S + S + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow} a + a + S \overset{\ell}{\Rightarrow} a + a + a$

Derivation for 2:
$S \Rightarrow S + \underline{S} \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$

Leftmost for 2:
$S \overset{\ell}{\Rightarrow} \underline{S} + S \overset{\ell}{\Rightarrow} a + S \overset{\ell}{\Rightarrow} a + S + S \overset{\ell}{\Rightarrow} a + a + S \overset{\ell}{\Rightarrow} a + a + a$

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 of this grammar does not have to be known for the exam

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

$aaabbb, ababab, aababb, \ldots$

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

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

Derivation for $aababb$:

$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow \ldots$ (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

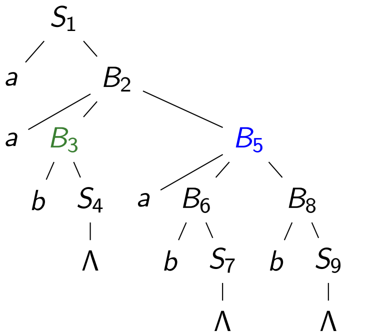
When a string has multiple variables, like $aabSB$ in the above example, then we are not forced to rewrite the first variable, we can as well rewrite another one.

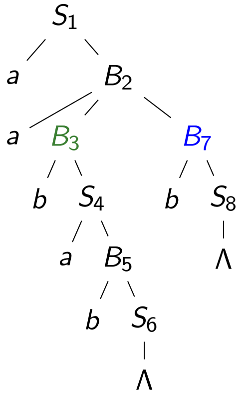
Thus we can do $aab\underline{S}B \Rightarrow aabB$, but also $aabS\underline{B} \Rightarrow aabSaBB$, for instance.

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.





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

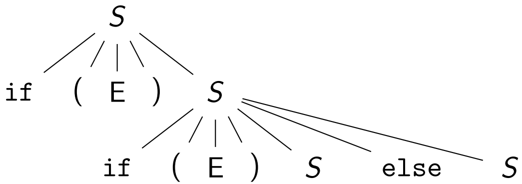
$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$

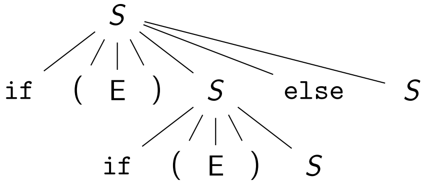
$S \to \texttt{if} \; (\; E \;) \; S \;\mid\; \texttt{if} \; (\; E \;) \; S \; \texttt{else} \; S \;\mid\; \ldots$

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

[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:

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

unambiguous...

[M] E 4.19

ambiguous:

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

unambiguous:

$S \to S_1 \mid S_2$

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

Some cf languages are inherently ambiguous

Ambiguity is *undecidable*

[M] Theorem 9.20

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	

just like FA, PDA accepts strings / language
just like FA, PDA has states
just like FA, PDA reads input one letter at a time
unlike FA, PDA has auxiliary memory: a stack
unlike FA, by default PDA is nondeterministic
unlike FA, by default Λ-transitions are allowed in PDA

Why a stack?

$$AnBn = \{a^i b^i \mid i \geq 0\}$$

with $x = aaabbb$

$$SimplePal = \{xcx^r \mid x \in \{a, b\}^*\}$$

with $x = aabcbaa$

Stack in PDA contains symbols from certain alphabet.
Usual stack operations: pop, top, push
Extra possiblity: replace top element X by string α

$AnBn = \{ \ a^n b^n \mid n \geq 0 \ \}$

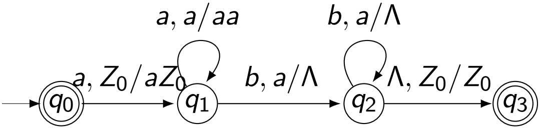
initial q_0, Z_0

PDA...

[M] E 5.3

$AnBn = \{ \ a^n b^n \mid n \geq 0 \ \}$

initial q_0, Z_0, accept $A = \{q_0, q_3\}$



[M] E 5.3