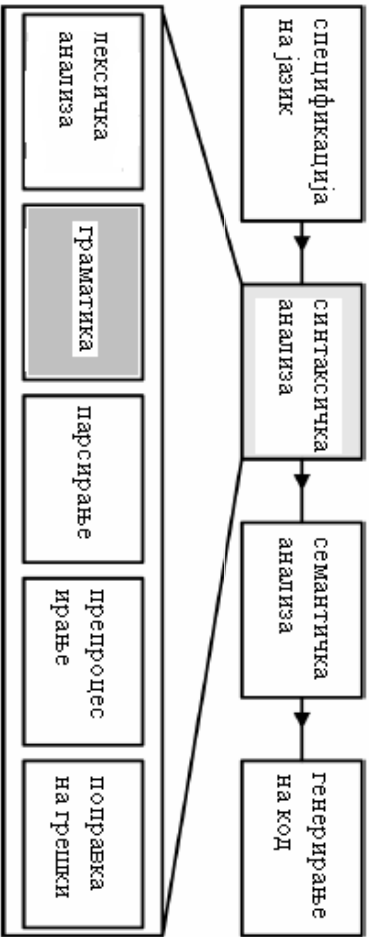


Граматикки



Јазици

- Природни јазици
- Компјутерски јазици

Јазици

- Регуларни јазици
- Контекстно слободни

Синтакса и семантика

- Пр. Прескокнува лисицата кучето над
 - Не ги следи правилата на граматиката на Македонскиот јазик-нема смисла
 - Синтаксички неточна
 - Секоја реченица треба да ги следи граматичките правила за да има значење
 - Но дали тоа е доволно?
-

Синтакса и семантика

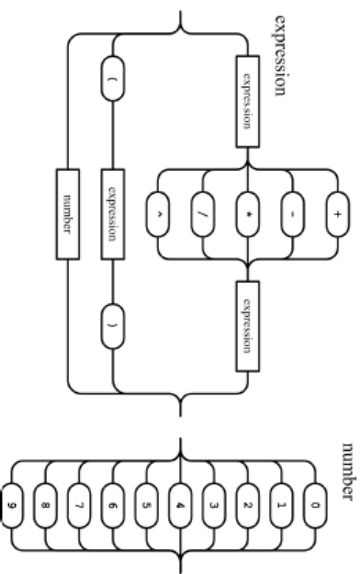
- Лисицата пие црвена боја
 - Точно наредени зборови според граматиката на македонскиот јазик-нема значење
 - Синтаксички точна-семантички неточна
-

Граматика

- Граматика е формален механизам за специфицирање на бесконечен јазик на конечен начин
- Граматиката се однесува само на синтаксата
- Правила на продукција

Граматика

- Проблеми
- Пр.



expression: expression + expression
| expression - expression
| expression * expression
| expression / expression
| expression ^ expression
| (expression)
| number
number: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.

Синтаксно дрво

■ Шема на лев извод за $1+2*3$

expression

\Rightarrow *expression* * *expression*

\Rightarrow *expression* + *expression* * *expression*

\Rightarrow *number* + *expression* * *expression*

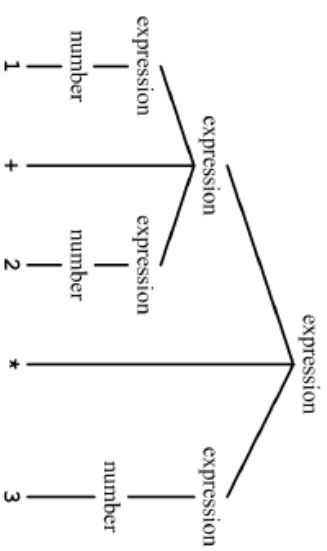
\Rightarrow $1 + \textit{expression} * \textit{expression}$

\Rightarrow $1 + \textit{number} * \textit{expression}$

\Rightarrow $1 + 2 * \textit{expression}$

\Rightarrow $1 + 2 * \textit{number}$

\Rightarrow $1 + 2 * 3$



Синтаксно дрво

■ Шема на десен извод за $1+2*3$

expression

\Rightarrow *expression* + *expression*

\Rightarrow *expression* + *expression* * *expression*

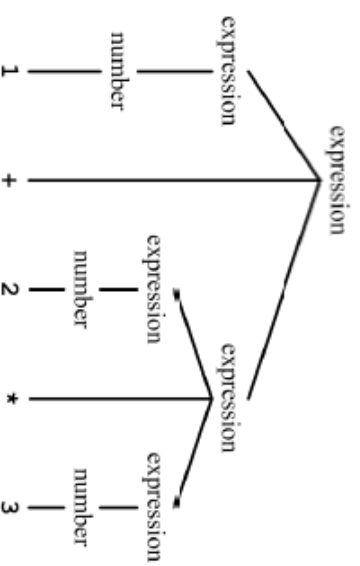
\Rightarrow *expression* + *expression* * *number*

\Rightarrow *expression* + *expression* * *3*

\Rightarrow *expression* + $2 * 3$

\Rightarrow *number* + $2 * 3$

\Rightarrow $1 + 2 * 3$



Двосмисленост

- Како да се реши проблемот за двосмисленост?
- Треба да се определи предност на операциите
- Решението лежи во додавање на нетерминалот

term

- Оригиналниот нетерминал *expression* сега се користи само за собирање и вадење
- Секогаш кога ќе се јави множење или делење парсирачкото дрво ќе содржи јазол *term* во кој ќе се извршат сите множење и делења пред да дојде до собирање или одземање

Недвосмислена граматика

expression :	term + expression
	term - expression
	term.
term :	factor * term
	factor / term
	factor ^ term
	factor.
factor :	0 1 2 3 4 5 6 7 8 9.
	(expression) .

\Rightarrow	<i>term</i> + expression
\Rightarrow	<i>factor</i> + expression
\Rightarrow	<i>1</i> + <i>expression</i>
\Rightarrow	<i>1</i> + <i>term</i>
\Rightarrow	<i>1</i> + <i>factor</i> * <i>term</i>
\Rightarrow	<i>1</i> + <i>2</i> * <i>term</i>
\Rightarrow	<i>1</i> + <i>2</i> * <i>factor</i>
\Rightarrow	<i>1</i> + <i>2</i> * <i>3</i>

Неавосмислена драматика

- Нетерминал фактор за замена на броевите и заградите кои имаат најголем приоритет.
- Колку подолу е еден терминал во граматиката, толку поголем приоритет има соодветната операција

<i>expression</i>	<i>expression</i>
\Rightarrow	\Rightarrow term + <i>expression</i>
\Rightarrow term + <i>expression</i>	\Rightarrow term + <i>term</i>
\Rightarrow factor + <i>expression</i>	\Rightarrow term + factor " * " term
\Rightarrow 1 + <i>expression</i>	\Rightarrow term + factor * <i>factor</i>
\Rightarrow 1 + <i>term</i>	\Rightarrow term + <i>factor</i> * 3
\Rightarrow 1 + <i>factor</i> * term	\Rightarrow term + 2 * 3
\Rightarrow 1 + 2 * <i>term</i>	\Rightarrow term + 2 * 3
\Rightarrow 1 + 2 * <i>factor</i>	\Rightarrow <i>factor</i> + 2 * 3
\Rightarrow 1 + 2 * 3	\Rightarrow 1 + 2 * 3

Неавосмислена праматика

<i>expression</i>	<i>expression(7)</i>
\Rightarrow	\swarrow
$term + expression$	$term(1)$
\Rightarrow	\swarrow
$factor + expression$	$expression(6)$
\Rightarrow	\swarrow
$1 + expression$	$term(6)$
\Rightarrow	\swarrow
$1 + term$	$term(3)$
\Rightarrow	\swarrow
$1 + factor * term$	$factor(2)$
\Rightarrow	\swarrow
$1 + 2 * term$	$term(3)$
\Rightarrow	\swarrow
$1 + 2 * factor$	$factor(3)$
\Rightarrow	\swarrow
$1 + 2 * 3$	1

Разрешување на проблемот со предност на операции

- ФОРТРАН

- + и – се заменуваат со))) + (((и))) - (((

- * и / се заменуваат со)) * ((и)) /((

- ** се заменуваат со) ** (

- Потоа се ставаат екстра \"(((\" на лево и \"))\" од десно

- Пр $(X + Y) + W/Z$ се добива

$$((((X))) + (((Y)))) + (((W))/((Z)))$$

Разрешување на проблемот со префиксна нотација

- Полскиот научник J. Łukasiewicz во доцните 20s.

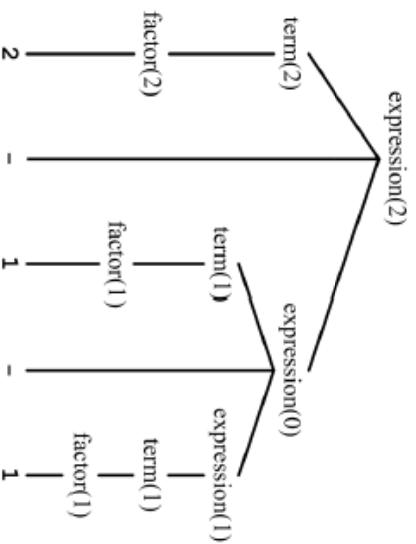
- Префиксна нотација, ослободување од загради или полска нотација

- Пример (Рех notation)

$$1 + 2 * 3 \text{ станува } + 1 * 2 3$$
$$1 / 2 - 3 \text{ станува } - / 1 2 3$$

Асоцијативност

- Пример: развивање на $2 - 1 - 1$ од претходната граматика



Асоцијативност

- Треба да се води сметка за асоцијативноста
- - и / се лево асоцијативни операции
- $^$ (степен), е десно асоцијативна операција
 $2^2^2 = 512 \neq 64$

Асоцијативност

expression: term + expression
 | term – expression
 | term.
term: factor * term
 | factor / term
 | factor ^ term
 | factor.
factor: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.
 | (expression) .

expression: expression + term
 | expression – term
 | term.
term: factor * term
 | factor / term
 | factor ^ term
 | factor.
factor: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.
 | (expression) .

Асоцијативност

expression: term + expression | term – expression | term.

- нетерминалот *expression* на левата страна на правилата се наоѓа на десната страна од знаците + и –
- Ова предизвикува десна рекурзија, бидејќи секој *expression* јазол повторно е *expression* јазол.
 - Лева рекурзија слишно може да се препознае
- Решението е да се воведе лева рекурзија за тие операции што се лево асоцијативни и десна за десно асоцијативните.

Граматика на Inger- BNF за логички јазик

statementlist :	statement statementlist.
statement :	identifier = implication ;.
implication :	conjunction restimplication .
restimplication :	ϵ .
restimplication :	\rightarrow conjunction restimplication.
restimplication :	\leftarrow conjunction restimplication.
restimplication :	\leftrightarrow conjunction restimplication.
conjunction :	negation restconjunction.
restconjunction :	ϵ .
restconjunction :	$\&$ negation restconjunction.
restconjunction :	negation restconjunction.
negation :	\sim negation.
negation :	factor .
factor :	(implication) .
factor :	1.
factor :	0.
factor :	A.
identifier :	...
identifier :	Z.

Граматика на Inger- EBNF за логички јазик

program :	{ statement ; } RESULT = implication.
statement :	identifier = implication.
implication :	conjunction { (\rightarrow \leftarrow \leftrightarrow) implication } .
conjunction :	negation { ($\&$) conjunction } .
negation :	{ \sim } factor .
factor :	(implication)
	identifier
	1
	0.
identifier :	A ... Z.

Listing 5.7: EBNF for Logic Language

『Dementia』