Структура на програма в C

Част 1

Иван Георгиев, Христо Иванов, Христо Стефанов

Технологическо училище "Електронни системи",
Технически университет, София

22 февруари 2019 г.

Съдържание

1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Съдържание

1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Синтаксис на програмен език

- Синтаксис на програмен език е набор от правила. Тези правила определят комбинациите от символи, които съставят валидно парче код на този език.
- Когато един компилатор компилира програма, той извършва *синтактичен анализ* на програмния код.
- Най-често преди извършването на синтактичния анализ се извършва *лексикален анализ*, който помага за по-лесното извършване на синтактичния анализ.
- След приключването на синтактичния анализ се извършва *семантичен анализ*, който проверява дали програмата е смислено построена.

Съдържание

1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Лексикален анализ

- *Лексикалният анализ* е етап, при който поредица от символи се превръща в поредица от лексеми.
- Лексемите са низове от символи с предварително зададено значение
- Броят, смисълът и начина на разпознаване на лексеми зависи от езика

Лексеми - пример

Примерен лексикален анализ на български език:

- Лексеми
  - *РАЗСТОЯНИЕ* - един или повече символи за празно място
  - *ДУМА* - един или повече символи от българската азбука
  - *ПУНКТУАЦИОНЕН ЗНАК* - един от символите - запетая, тире, точка, точка и запетая, две точки.
- Редица от символи:
  - *Иван дойде по-късно днес.*
- Резултат
  - [*ДУМА, РАЗСТОЯНИЕ, ДУМА, РАЗСТОЯНИЕ, ДУМА, ПУНКТУАЦИОНЕН ЗНАК, ДУМА, РАЗСТОЯНИЕ, ДУМА, ПУНКТУАЦИОНЕН ЗНАК*]

Съдържание

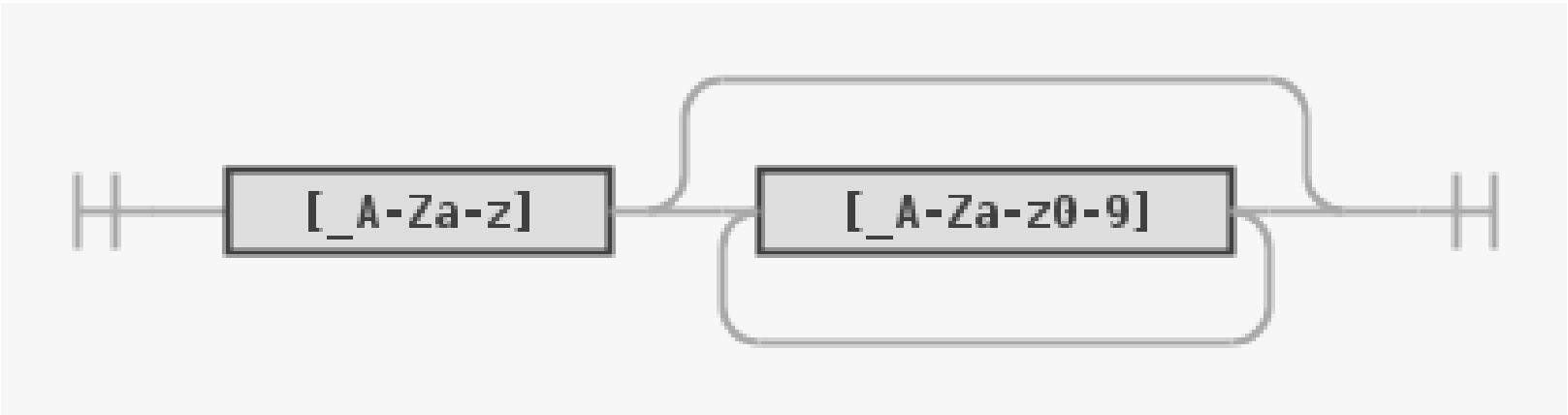
1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Лексеми в езика C

- Основните лексеми в езика C са:
  - идентификатор (*identifier*)
  - константа (*constant*)
  - низов литерал (*string literal*)
  - пунктуатор (*punctuator*)
  - коментар (*comment*)
  - празно място (*white space*)

Идентификатори

- *Идентификатор* - някой от символите - ’_’, ’A’ до ’Z’, ’a’ до ’z’ - следван от нула или повече от следните символи - ’_’, ’A’ до ’Z’, ’a’ до ’z’, ’0’ до ’9’
- Примери:
  - a, A, b, _, HELLo, _hello, ___hello, hello1234, hello_world



Фигура 1: Диаграма

Ключови думи

- *Ключовите думи* са лексеми, които отговарят на описанието за идентификатори, но имат различен смисъл в езика от този на идентификаторите.
- Списък на ключовите думи в C:

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Константи

- *Константа* - цяло число в десетичен, шестнадесетичен, осмичен или символен запис[1] или рационално число в десетичен или шестнадесетичен запис
- Примери:
  - Цели числа
    - Десетичен запис - `1, 2, 2019`
    - `Шестнадесетичен запис - 0x1f, 0x003, 0xabcdef, 0xABCDEF`
    - `Осмичен запис - 0123, 0777`
    - `Символен запис - 'j', '0', '\0', '\n', '\152', '\x6A'`
  - Рационални числа
    - Десетичен запис - `1.0, 3.14`
    - `Шестнадесетичен запис - 0x1f.3bp0`

[1] Обяснение на следващия слайд

Символен запис

- *Символният запис* на цели числа е символ ограден от единични кавички - ’
- Стойността на числото записано чрез символен запис се определя от номера в ASCII таблицата[2] на ограждения символ
- Примери
  - ’a’, ’b’, ’c’, ’!’, ’?’, ’_’ $\rightarrow$ 97, 98, 99, 33, 63, 95
- За символите от ASCII таблицата, които нямат определен графичен символ (*non-printable characters*, например символ за нов ред, нулев символ) се налага използването на *специални последователности*
- Когато ограденият символ е ’ също се налага използването на специална последователност

[2] http://www.asciitable.com/

Специални последователности от символи

- *Специалните последователности* (*escape sequences*) са последователности от символи, които се интерпретират като един символ от ASCII таблицата.

Последователност	Наименование	Номер в ASCII
\a	alarm or beep	7
\b	backspace	8
\f	form feed	12
\n	new line	10
\r	carriage return	13
\t	tab (horizontal)	9
\v	vertical tab	11
\\	backslash	92
\'	single quote	27
\"	double quote	22
\?	question mark	63
\nnn	octal number	nnn
\xhh	hexadecimal number	hh
\0	null	0

Специални последователности от символи - примери

- Символни записи използващи специални последователности
  - `'\n', '\t'`
  - `'\'', '\\'`
  - `'\047', '\092'`
  - `'\x27', '\x5c'`

Низов литерал

- *Низов литерал* (*string literal*) - низ от символи оградени с двойни кавички - '"'. Може да съдържа специални последователности от символи.
- Примери:
  - "Hello world"
  - "asdf"
  - "First line\nSecond\tline"
  - "String literal with \"quotes\" is also possible"
  - "Single quotes don't need escape sequences in string literals"

Пунктуатори

- *Пунктуатор* (*punctuator*) - някоя от следните комбинации от пунктуационни символи:

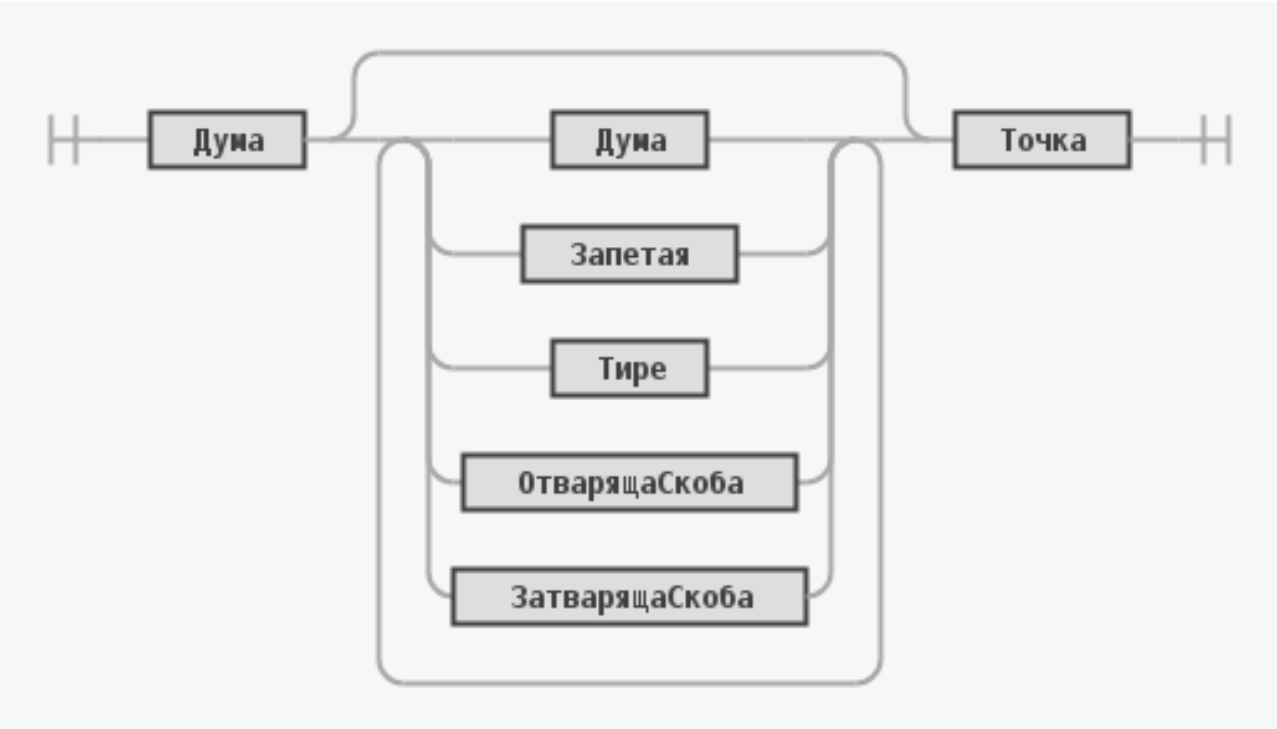
```
[     ]     (     )     {     }     .     ->
&     *     +     -     ~     !
++    --    /     ?     =     ,
%     <<    >>    <     >     <=    >=
:     ;     ...
*=    /=    %=    +=    -=    <<=
==    >>=   !=    &=    ^
|     ^=    &&    ||    |=
```

Съдържание

1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Синтактичен анализ и правила върху лексеми

- *Синтактичният анализ* е етап, при който поредицата от лексеми, получена от лексикалния анализ, се проверява дали отговаря на синтактичните правила на езика
- Пример:

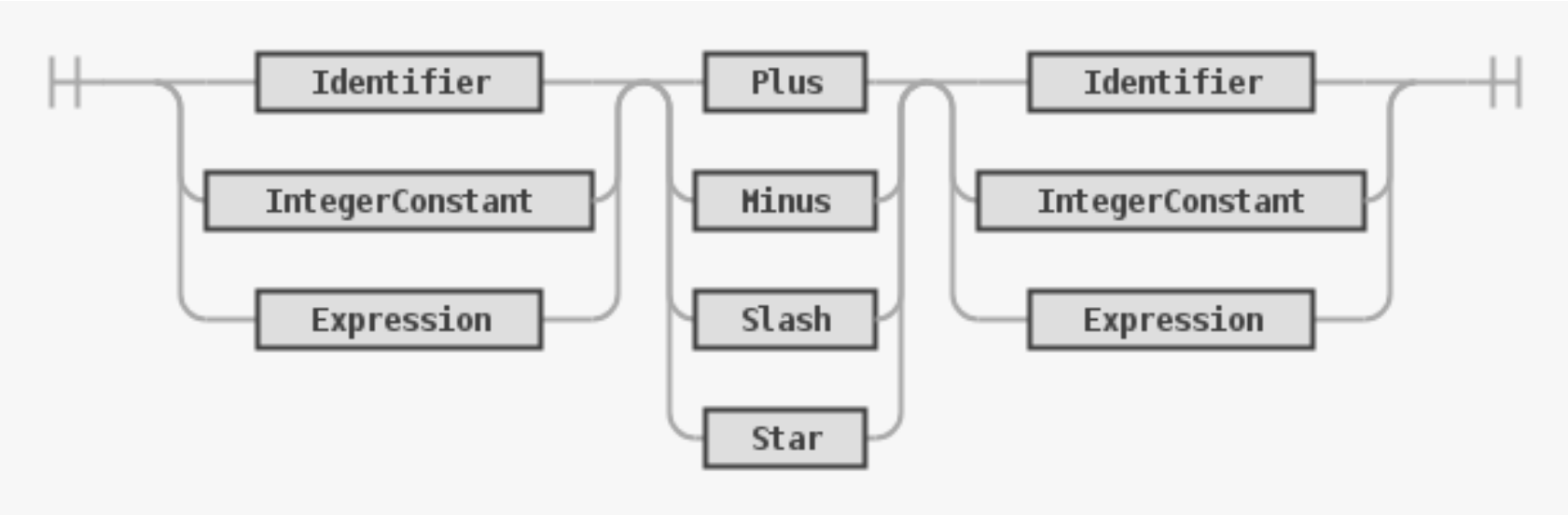


Фигура 2: Непълна диаграма за изречение на български език

Съдържание

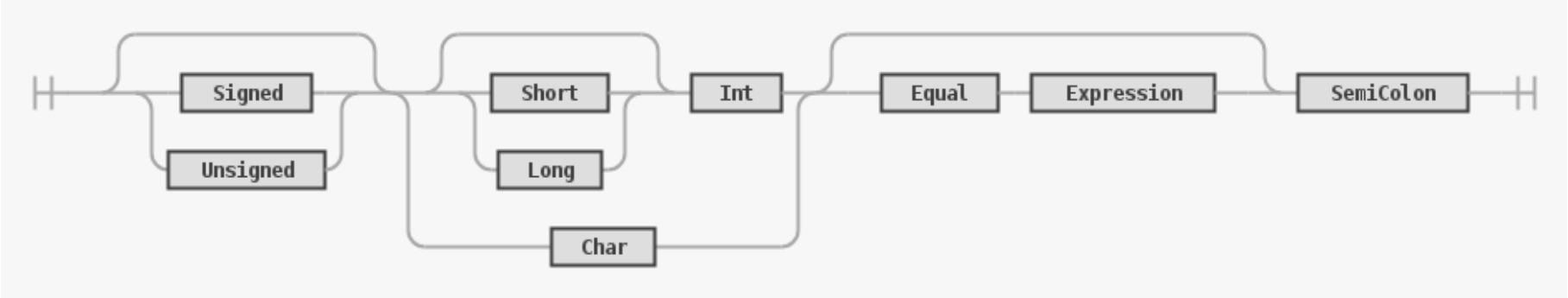
1. Синтаксис на програмен език
2. Лексикален анализ и лексеми
3. Лексеми в езика C
4. Синтактичен анализ и правила върху лексеми
5. Синтактичен анализ в C

Синтаксис на аритметичен израз



Фигура 3: Непълна диаграма за аритметичен израз в C

Синтаксис на декларация на променлива



Фигура 4: Непълна диаграма за декларация на променлива в C